
FindFace

Выпуск 2.0

NtechLab

мар. 29, 2024

1	Новое в FindFace Multi 2.0	5
2	Руководство системного администратора	9
2.1	Архитектура	9
2.2	Требования	16
2.3	Развертывание и удаление FindFace Multi	18
2.4	Администрирование и основная конфигурация	62
2.5	Настройка нейронных сетей FindFace Multi	117
2.6	Обслуживание и устранение неисправностей	161
2.7	Приложения	195
3	Руководство пользователя	197
3.1	Приступая к работе	197
3.2	Основы веб-интерфейса	200
3.3	Картотека	201
3.4	Источники видео	205
3.5	События и эпизоды распознавания объектов	218
3.6	Поиск объектов в системе	230
3.7	Сравнение двух объектов	231
3.8	Счетчики лиц, силуэтов и транспортных средств. Определение дистанции	232
3.9	Кластеры лиц, силуэтов, транспортных средств	238
3.10	Отчеты	248
3.11	Журнал действий пользователей	251
3.12	Связанная с людьми аналитика	252
3.13	Видеонаблюдение	255
4	Руководство по интеграции	259
4.1	HTTP API	259
4.2	Вебхуки	300
4.3	Внешние системы управления видео (VMS)	311
4.4	Внешние детекторы	318

FindFace Multi — это многофункциональная мультиобъектная система видеоаналитики, в основе которой лежит [FindFace Enterprise Server](#), передовая технология распознавания на базе искусственного интеллекта. FindFace Multi представляет собой готовый к использованию продукт, который может использоваться в таких областях, как транспорт, розничная торговля, банковское обслуживание, индустрия развлечений, спортивные мероприятия, организация мероприятий, сервисы знакомств, видеонаблюдение, общественная и корпоративная безопасность и др.

FindFace Multi может обнаруживать, идентифицировать и анализировать следующие объекты на видеоизображении:

- Человеческие лица, вместе с распознаванием таких атрибутов, как пол, возраст, эмоции, наличие очков, бороды, медицинской маски и многих других. Интегрированная в FindFace Multi антиспуфинговая система гарантирует, что перед камерой находится живой человек, и исключает возможность мошенничества с использованием фотографии лица на бумаге или экране мобильного устройства.
- Человеческие силуэты, вместе с распознаванием типа и цвета одежды.
- Транспортные средства, вместе с распознаванием таких атрибутов, как марка, модель, тип кузова, цвет, номер и других.

За доли секунды после идентификации и анализа объекта FindFace Multi уведомляет ответственных лиц о его появлении. Дополнительная информация об объекте, такая как пол, возраст человека, ГРЗ транспортного средства и др., отображается в соответствующем событии распознавания.

FindFace Multi поддерживает интеграцию сторонних решений через [HTTP API](#) и [вебхуки](#), так что вы с легкостью сможете усовершенствовать свою текущую систему или приложение, добавив в них функциональность распознавания объектов.

Ключевые функции

- Платформа на основе AI.
- Комплексная картотека, содержащая сборную информацию о человеке/транспортном средстве.
- Быстрый и надежный мониторинг видео в реальном времени по картотеке.
- Мультиобъектная идентификация и аналитика: лица, силуэты, транспортные средства.
- Поддержка потокового видео и архивов, большинства видеоформатов и кодеков, которые могут быть декодированы FFmpeg.
- Прогрессивное управление камерами, включая поддержку ONVIF, возможность изменения ориентации видео, точной настройки камеры под распознавание объектов каждого типа, видеозаписи (если активирована функция Видеомэгнитофон).
- Мультиобъектная верификация: лица, силуэты, транспортные средства.
- AI-распознавание пола, возраста, эмоций, очков, бороды, медицинской маски и других атрибутов лица.
- AI-детектор живых лиц (Liveness).
- AI-распознавание конкретных персон и транспортных средств.
- AI-распознавание типа и цвета одежды.
- AI-распознавание специальных транспортных средств, марки, модели, типа кузова, цвета, номера ТС.
- Поиск лиц, силуэтов, транспортных средств в базах данных.

- Возможность подсчета лиц, силуэтов и транспортных средств на подключенных камерах и определения расстояния между силуэтами. Поддержка подсчета по одной и нескольким камерам.
- Видеонаблюдение. Запись видео с камер (Видеомагнитофон).
- Возможность отслеживания присутствия людей в определенных областях по заданным правилам и расписаниям мониторинга.
- Автоматическая кластеризация объектов одного происхождения (изображения лица/силуэта, принадлежащие одному человеку; изображения одного и того же транспортного средства), обогащенная сквозной интеграцией с картотекой.

Среда развертывания

- Удобный консольный инсталлятор и дружелюбный интерфейс.
- Развертывание на одном или нескольких серверах.
- Повышенная производительность и отказоустойчивость в высоконагруженных системах с большим количеством подключенных камер и клиентов.
- Возможность лицензирования в открытых и закрытых системах.
- CPU- и GPU-ускорение на ваш выбор.

Безопасность системы

- Расширенное управление пользователями.
- Аутентификация на основании пароля, сертификата и лица для гарантированной защиты системы.
- Безопасность данных в картотеке.
- Комплексный, удобный журнал действий пользователя с возможностью поиска.
- Утилиты резервного копирования и восстановления.
- Возможность мониторинга сессий пользователей и блокировки устройств без деактивации учетных записей пользователей.

Этика использования данных

- Полная поддержка законов о защите персональных данных (GDPR и аналогичных).

Расширенное использование системы

- Анализ социальных взаимодействий.
- Анализ аудитории.
- Детализированные отчеты на события распознавания лиц, эпизоды, события поиска, кластеры, камеры, карточки, журнал действий пользователей и аудиторию.
- Детектор живых лиц (liveness) как автономный сервис.

Полезные мелочи

- Быстрое создание картотеки.
- Полная настройка картотеки.
- Поддержка дедупликации событий и карточек.
- Расширенный набор поисковых фильтров.
- Очистка базы данных по расписанию.

Интеграция

- Интеграция через HTTP API и вебхуки.
- Партнерские интеграции с популярными системами.
- Интеграция с периферийными устройствами.

Новое в FindFace Multi 2.0

Улучшенные алгоритмы, интерфейс, удобство использования:

- Совершенно новый дизайн для быстрой и комфортной работы.

Полностью обновленный пользовательский интерфейс обеспечивает плавную работу с большими объемами данных: просмотр подробной информации об объекте на боковой панели, легкое переключение между открытыми вкладками. Начните изучать новый интерфейс с [основ](#).

- Лучшие нейронные сети: технология Liveness, получившая сертификацию iBeta Level 2.

В этой версии FindFace Multi представлена совершенно новая модель нейронной сети для мобильных телефонов `liveness.goodwin`, получившая сертификацию iBeta Level 2. Модель предназначена для обнаружения атак с помощью силиконовых/латексных масок и бумажных масок, воспроизводящих текстуру лица. Кроме того, FindFace Multi 2.0 включает в себя огромное количество совершенно новых моделей нейронных сетей или обновленных версий предыдущих моделей нейронных сетей с улучшенными характеристиками.

См. [Настройка нейронных сетей FindFace Multi](#)

Технические изменения:

- Техническая архитектура: микросервисы вместо монолита.

См. [Архитектура](#)

- Распространение на основе Docker: FindFace Multi 2.0 работает как набор контейнеров Docker, описанных в файле `docker-compose`.

См.:

- [Платформа на основе Docker](#)
- [Полезные команды Docker](#)

- Поддержка вертикального масштабирования: теперь можно обрабатывать больше событий благодаря более эффективному использованию мощности сервера.

- Поддержка горизонтального масштабирования: за счет развертывания новых серверов можно обрабатывать большее количество событий за меньшее время.

Поддержка вертикального и горизонтального масштабирования ведет к повышению производительности.

- Поддержка *S3-совместимых хранилищ*: S3 хранилище обеспечивает надежное и долгосрочное хранение неограниченного количества файлов и данных. Это дает возможность избежать ограничений файловой системы, когда речь идет о больших объемах данных.
- Поддержка операционной системы CentOS.

Новые функции:

- *Внешние системы управления видео (VMS)*.
- Управление PTZ-камерами: в панель управления видеоплеера добавлена возможность поворота для PTZ-камер по ONVIF.
- *Умная очистка видеоархива*: настройте регулярную очистку видео.
- Расширенные *вебхуки*.
- Интеграция с *Active Directory*.
- *Фильтры с полным набором атрибутов для ленты событий и эпизодов на Видеостене*.
- *Отображение эпизодов на временной шкале в Видеоплеере*.
- Распознавание новых атрибутов объекта.

См.:

- *Распознавание лиц и их атрибутов*
- *Распознавание транспортных средств и их атрибутов*
- *Распознавание спецтранспорта*
- *Распознавание силуэтов и их атрибутов*
- Новый подход к лицензированию атрибутов: некоторые атрибуты, которые ранее не лицензировались, или лицензировались как группа, объединяющая сразу несколько атрибутов объекта, теперь лицензируются отдельно.

Переименования:

- *Камеры, Видео* → *Источники видео*; *Эпизоды, События* → *Эпизоды и события*; *Аналитика* → *Анализ аудитории*; *Карточки* → *Картотека*.
- *автомобиль* → *транспортное средство*, *ТС*

Новые нейронные сети:

• Обнаружение объектов

- Обнаружение лиц: новая модель для обнаружения лиц `face.jasmine_fast.003` с улучшенными характеристиками.
- Обнаружение транспортных средств: совершенно новые модели нейронных сетей для обнаружения транспортных средств: `car.gustav_accurate.004` (включена в `findface-extraction-api`) и `car.jasmine_fast.005` (включена в `findface-video-worker`).
- Обнаружение силуэтов: совершенно новые модели нейронных сетей для обнаружения силуэтов: `body.gustav_normal.015` (включена в `findface-extraction-api`) и `body.jasmine_fast.018` (включена в `findface-video-worker`).
- Детектор 3-в-1: новый детектор `headbodyface.alpha000_normal.001`, который возвращает три подобъекта, принадлежащих человеку: силуэт/голова/лицо.

• Нормализация изображений объектов

- Нормализация изображения лица: новые модели нормализации изображений лиц `facenorm.multicrop_full_center_size400` и `facenorm.multicrop_full_crop2x_size400`, использующие усовершенствованный алгоритм нормализации, и новая версия `bee.v3` с улучшенными характеристиками.
- Нормализация изображения регистрационного знака автомобиля: новая и более быстрая модель `briacpn.v0`, а также новые версии `anaferon.v5` и `anaferon.v7` с улучшенными характеристиками.

• Распознавание объектов

- Распознавание лиц: нейросети `lime.v2` и `mango_320`, обеспечивающие быстрое распознавание лиц.
- Распознавание силуэтов: новая модель для распознавания силуэтов `pedrec.clio` с повышенной точностью распознавания.

• Распознавание атрибутов объектов

- Распознавание положения головы: новая нейронная сеть `headpose.v2` для распознавания наклона/поворота головы.
- Распознавание витальности: новая модель для мобильных телефонов `liveness.goodwin`, успешно прошедшая сертификацию iBeta Level 2. Модель предназначена для обнаружения атак с помощью силиконовых/латексных масок и бумажных масок, воспроизводящих текстуру лица. Совершенно новая модель нейронной сети `liveness.web.v0` для обнаружения спуфинг-атак на веб-камеры. Новая версия `liveness.pacs.v2` с улучшенными характеристиками.
- Распознавание качества изображения лица: новая модель `quality_fast.v1`, которая предсказывает качество изображения лица для выбора лучшего кадра и фильтрации мусора.
- Распознавание категорий транспортных средств: новая нейронная сеть `carattr.categories.v0`, которая определяет, принадлежит ли транспортное средство к одной из этих категорий: мотоцикл (включая мопед и скутер) или квадроцикл, легковой автомобиль, грузовик, легковой автомобиль с прицепом, грузовик с прицепом, автобус или сочлененный автобус.
- Распознавание транспортных средств по весу и размеру кузова: новая нейронная сеть `carattr.weight_types7.v0`, которая предсказывает класс транспортного средства в зависимости от его веса и размера кузова.

- Распознавание ракурса ТС: новая нейронная сеть `carattr.orientation.v0` для распознавания ракурса транспортного средства: вида спереди, сзади или сбоку.
- Распознавание спецтранспорта: новая версия `carattr.special_types11.v1`, поддерживающая распознавание маршрутного транспорта, автомобилей каршеринга, газовых служб, военной техники, дорожных служб, а также автомобилей полиции, скорой помощи, спасательных служб и такси.
- Распознавание регистрационных знаков транспортных средств: новая версия `carattr.license_plate.v7`, которая поддерживает распознавание номеров Чехии, Пакистана, Таиланда, а также уже включенных в версии `carattr.license_plate.v6` и `carattr.license_plate.v5` номеров Сербии, Литвы, Латвии, Молдовы, Эстонии, Финляндии, Азербайджана, Таджикистана, Туркменистана, Мексики и Аргентины. Поддерживается распознавание регистрационных знаков транспортных средств ОАЭ, России, Казахстана, Грузии, Саудовской Аравии, Бразилии, Индии, Узбекистана, Вьетнама, Белоруссии, Украины, Армении и Кыргызстана, представленных в более ранних версиях продукта – в сумме поддерживается 27 стран.
- Качество изображения регистрационного знака: новая и гораздо более быстрая версия `carattr.license_plate_quality.v1`, которая используется для получения наилучшего кадра регистрационного знака ТС из серии снимков.
- Распознавание возраста и пола по силуэту: новая нейронная сеть `pedattr.age_gender.v0` для предсказания возраста и пола человека по силуэту.
- Распознавание наличия сумки: новая нейронная сеть `pedattr.bags.v0`, предсказывающая есть ли у человека сумка, рюкзак или чемодан.
- Распознавание СИЗ: новая нейронная сеть `pedattr.protective.v1` для распознавания наличия или отсутствия средств индивидуальной защиты и их цвета.

Изменения функциональности по сравнению с версией 1.2:

- Области убраны из веб-интерфейса, но их функциональность по-прежнему доступна через API. В будущем функциональность Областей будет включена в Счетчики. Рассмотрите возможность использования предыдущей версии FindFace Multi, если вы активный пользователь Областей.
- График счетчика за последний час/день/неделю удален из интерфейса настроек счетчика: график не подходил для анализа данных. Анализ данных, полученных от счетчиков, доступен через выгрузку отчетов или постобработку данных из API.
- Из интерфейса убраны Связи между карточками, но эта функция все еще существует в API: будучи экспериментальной, функция Связей между карточками не смогла закрыть потребности потенциальных клиентов. Возможно, эта функциональность будет доработана в будущем.

Руководство системного администратора

Эта глава посвящена разворачиванию и последующему обновлению и техническому обслуживанию FindFace во время эксплуатации.

2.1 Архитектура

Хотя взаимодействие с FindFace Multi происходит в основном через веб-интерфейс, не забудьте уделить немного времени изучению архитектуры программного комплекса. Эти знания необходимы для разворачивания, интеграции, обслуживания и устранения проблем при работе FindFace Multi.

В этой главе:

- *Объекты распознавания. Процесс распознавания*
- *Платформа на основе Docker*
- *Архитектурные элементы*
 - *Схема архитектуры*
 - *Ядро FindFace*
 - *Прикладной модуль FindFace Multi*
 - *Видеомагнитофон*
- *Разворачивание на одиночном сервере или в кластере*
- *Аппаратное ускорение на CPU и GPU*

2.1.1 Объекты распознавания. Процесс распознавания

FindFace Multi может распознавать следующие объекты и их атрибуты:

- человеческие лица
- человеческие силуэты
- транспортные средства

FindFace Multi обнаруживает объект на фотографии или видеоизображении и подготавливает его изображение с помощью нормализации. Далее нормализованное изображение используется для извлечения вектора признаков объекта (n-мерного вектора числовых признаков, которые характеризуют объект). Векторы признаков объектов хранятся в базе данных и в дальнейшем используются для верификации и идентификации.

2.1.2 Платформа на основе Docker

FindFace Multi разворачивается в Docker, наборе продуктов формата платформа как услуга (PaaS), которые используют виртуализацию на уровне операционной системы, чтобы поставлять программное обеспечение в пакетах, называемых контейнерами. Каждый сервис FindFace Multi работает в контейнере Docker.

2.1.3 Архитектурные элементы

FindFace Multi состоит из следующих основных архитектурных элементов:

- Ядро FindFace, передовая технология распознавания объектов на базе искусственного интеллекта, которая может использоваться в качестве отдельного продукта [FindFace Enterprise Server](#).
- FindFace Multi, готовый прикладной модуль к FindFace Enterprise Server.

Внутренняя архитектура Ядра FindFace позволяет встроить дополнительную функциональность «Видеомониторинг», которая позволяет записывать, хранить и проигрывать видеоданные с камер.



Ядро FindFace

Ядро FindFace включает в себя следующие компоненты:

2.1. Архитектура

Прикладной модуль FindFace Multi

Прикладной модуль FindFace Multi включает в себя следующие компоненты:

Ком- по- нент	Ис- поль- зу- е- мые пор- ты	Описание	По- став- щик
findface-multi-legacy	Настраивается	Сервис, который служит шлюзом доступа к ядру FindFace. Обеспечивает взаимодействие между ядром FindFace и веб-интерфейсом, работой системы в целом, HTTP и веб-сокетом, мониторингом объектов, уведомлениями о событиях, эпизодами, вебхуками, кластеризацией объектов и счетчиками.	Собственная разработка NtechLab
findface-multi-pause	н/п	Внутренние сервисы, помогающие findface-multi-legacy . Сервис findface-multi-audit является заделом на будущее. Он станет полностью функциональными в следующих версиях. На данный момент он частично дублирует функциональность findface-multi-legacy .	NtechLab
findface-multi-audit	8012, 8013, 8014	Используйте findface-multi-legacy для работы с ним. Сервис findface-multi-identity-provider служит для аутентификации и управления пользователями и ролевой моделью доступов.	
findface-multi-identity-provider	8022, 8023, 8024		
findface-onvif-discovery	н/п		
findface-multi-ui	Настраивается	Основной веб-интерфейс, использующийся для взаимодействия с FindFace Multi. Функционирует на основе фреймворка Django. Позволяет работать с событиями распознавания объектов, искать объекты в базах данных, управлять камерами, пользователями, карточками и списками наблюдения, собирать статистику в реальном времени и многое другое.	
NATS	4222	Стороннее программное обеспечение, которое реализует брокер сообщений внутри findface-multi-legacy .	NATS
etcd	2379	Стороннее программное обеспечение, реализующее блокировки в сервисе findface-multi-legacy , например блокировки в механизме проверки лицензии, отчетах, обработке видео, кластеризации объектов и т. д.	etcd
Pgbouncer	5439	Стороннее программное обеспечение, обеспечивающее облегченный пул соединений для PostgreSQL. Необязательный компонент, используется для увеличения производительности базы данных при высокой нагрузке.	PgBouncer
PostgreSQL	5432	Стороннее программное обеспечение, реализующее основную базу данных системы. В базе данных хранятся карточки людей и транспортных средств, а также данные для внутреннего использования, включая учетные записи пользователей и настройки камер. Векторы признаков объектов и события распознавания объектов хранятся в Tarantool (часть ядра FindFace).	PostgreSQL

Видеомагнитофон

Видеомагнитофон является дополнительно устанавливаемой частью ядра FindFace. Видеомагнитофон включает в себя следующие компоненты:

Ком- по- нент	Ис- поль- зу- е- мые пор- ты	Описание	По- став- щик
findface- video- storage	8611	Сервис, реализующий управление видеофрагментами. Получает видеофрагменты от компонента findface-video-worker , помещает их в хранилище (findface-upload), записывает метаинформацию и сведения о местонахождении видеофрагментов в базу данных Видеомагнитофона (MongoDB). По запросу от findface-multi-legacy выдает информацию об имеющихся видеофрагментах в форме ссылок на Websocket-поток. Сервис findface-video-streamer использует данные ссылки, чтобы передать видео пользователю для просмотра и скачивания.	Соб- ствен- ная раз- ра- бот- ка NtechLab
findface- video- streamer	9000	По запросу от findface-multi-ui данный сервис по Websocket извлекает требуемые видеофрагменты из хранилища findface-video-storage и findface-video-worker (только последний фрагмент, если он еще не записан в хранилище). Далее он склеивает видеофрагменты в единое видео и передает его пользователю для просмотра и скачивания.	
MongoDB	27017	Стороннее программное обеспечение, реализующее базу данных Видеомагнитофона. База данных хранит метаинформацию о видеофрагментах, включая сведения об их местонахождении. Сами видеофрагменты хранятся в компоненте findface-upload .	MongoDB

См. также:

- *Хранилища данных FindFace Multi*

2.1.4 Развертывание на одиночном сервере или в кластере

Вы можете развернуть FindFace Multi как на одиночном сервере, так и в многосерверной среде. При выборе последнего варианта доступны следующие схемы развертывания:

- Центральный сервер FindFace Multi, взаимодействующий с несколькими дополнительными серверами для обработки видео (серверами, на которых установлен только **findface-video-worker**).

См. раздел *Развертывание FindFace Video Worker на удаленных серверах*.

- Полностью распределенная архитектура FindFace Multi. Может понадобиться балансировка нагрузки. Свяжитесь с нашими экспертами по адресу support@ntechlab.com для получения рекомендаций.

См. раздел *Стандартное развертывание в многосерверной среде*.

2.1.5 Аппаратное ускорение на CPU и GPU

Сервисы `findface-extraction-api` и `findface-video-worker` могут использовать как CPU-, так и GPU-ускорение. Нужный тип ускорения выбирается во время установки из консольного инсталлятора.

Важно: Для выбора конфигурации оборудования см. *Требования*.

Важно: Если разрешение используемой камеры превышает 1280x720 пикселей, настоятельно рекомендуется использовать пакет `findface-video-worker` с ускорением на GPU.

Примечание: *Liveness-детектор* на CPU работает гораздо медленнее, чем на GPU.

2.2 Требования

В этой главе:

- Системные требования для базовой конфигурации
- Необходимые навыки администратора
- Форматы видеофайлов

2.2.1 Системные требования для базовой конфигурации

Для расчета характеристик серверов FindFace используйте приведенные ниже требования.

Совет: Сначала обязательно ознакомьтесь с *архитектурой* FindFace.

Важно: Если разрешение видео превышает 1280x720 пикселей, настоятельно рекомендуется использовать пакет с ускорением на GPU `findface-video-worker-gpu`.

Важно: На серверах с процессором AMD полная функциональность CPU-сервиса `findface-extraction-api` не гарантируется. Вместо него используйте GPU-сервис `findface-extract-api-gpu` вместе с версией нейронных сетей для GPU.

Примечание: В случае высоконагруженной системы рекомендуется использовать SSD.

	Минимальная	Рекомендуемая
CPU	Intel Core i5 CPU с 4+ физическими ядрами 3+ ГГц. Поддержка AVX2	Intel Xeon Silver/Gold с 6+ физическими ядрами
	На собственные нужды FindFace требуется 2 ядра HT > 2.5 ГГц. Характеристики также зависят от количества одновременно обрабатываемых видеофайлов. Для одного файла 720p@25FPS требуется 2 ядра >2.5 ГГц. Поддержка AVX2.	
GPU (опционально)	NVIDIA GeForce® GTX 1060 6 Гб	NVIDIA GeForce® GTX 1080Ti+ с 11+ Гб RAM
	Поддерживаемые устройства: NVIDIA, архитектура Pascal и выше. <i>Примечание: видеокарты NVIDIA GeForce RTX 40 Series в настоящее время не поддерживаются.</i>	
RAM	10 Гб	16+ Гб
	На собственные нужды FindFace требуется 8 Гб. Потребление памяти также зависит от количества обрабатываемых видеофайлов. Для одного видеофайла 720p@25FPS требуется 2 Гб RAM.	
HDD (SSD для лучшей производительности)	16 Гб	16+ Гб
	На собственные нужды операционной системы и FindFace требуется 15 Гб. Суммарный объем определяется в зависимости от требуемой глубины архива событий в базе данных и в логе из расчета 1.5 Мб на 1 событие.	
Операционная система	Ubuntu от 18 до 22, только x64, RHEL / CentOS 7, Debian 11.	

Примечание: Вы также можете использовать виртуальную машину на базе процессора Intel, если поддерживаются инструкции AVX2, а 8 физических ядер выделены исключительно виртуальной машине.

Совет: Для более точного подбора конфигурации свяжитесь с нашими техническими экспертами по адресу support@ntechlab.com.

2.2.2 Необходимые навыки администратора

Администратор FindFace Multi должен обладать навыками работы с ОС, на которой развернут экземпляр продукта, на уровне продвинутого пользователя.

2.2.3 Форматы видеофайлов

FindFace Multi поддерживает широкий спектр форматов файлов в зависимости от типа ускорения, CPU или GPU.

Инсталляции с ускорением CPU и GPU поддерживают все кодеки FFmpeg. Кроме того, поддерживаются следующие кодеки:

- *CPU-ускорение:* FLV (как кодек, так и как контейнер), H263, H264, H265, MJPEG, VP8, VP9, MPEG1VIDEO, MPEG2VIDEO, MSMPEG4v2, MSMPEG4v3.
- *GPU-ускорение:* MJPEG, H264, H265, VP9 и другие, в зависимости от списка кодеков, поддерживаемых используемой видеокартой. Кроме того, для инсталляций с `video-worker-gpu`, можно

расширить количество поддерживаемых кодеков, включив декодирование видео на CPU, которое недоступно по умолчанию.

Чтобы включить декодирование видео на CPU для инсталляций на GPU-ускорении, выполните следующее:

1. Откройте файл `/opt/findface-multi/configs/findface-video-worker/findface-video-worker.yaml`.

```
sudo vi /opt/findface-multi/configs/findface-video-worker/findface-video-worker.yaml
```

2. Установите `cpu: true` в разделе `video_decoder`.

```
...
video_decoder:
  cpu: true
...
```

3. Перезапустите контейнер `findface-multi-findface-video-worker-1`.

```
sudo docker container restart findface-multi-findface-video-worker-1
```

2.3 Развертывание и удаление FindFace Multi

Платформа на основе Docker

FindFace Multi 2.0 использует для работы платформу Docker. Прежде чем приступить к развертыванию FindFace Multi, вам потребуется установить и запустить продукты Docker. Для вашего удобства данная глава содержит раздел *Подготовка сервера Ubuntu*, описывающий тонкости установки Docker в Ubuntu. Для других платформ см. документацию [Docker](#).

Драйвер NVIDIA и NVIDIA Container Runtime (только для GPU)

Если вы собираетесь развернуть FindFace Multi с GPU-ускорением, вам потребуется установить драйвер NVIDIA и среду NVIDIA Container Runtime. Информацию об этом вы найдете в разделе *Подготовка сервера Ubuntu*.

Варианты развертывания

После завершения подготовки сервера настало время развернуть FindFace Multi. Здесь предоставляются следующие возможности:

1. Автоматическая установка односерверной конфигурации FindFace Multi. Будучи самым простым, данный тип установки отлично подходит для начала работы с FindFace Multi. Мы рекомендуем выбрать его, если вы только знакомитесь с системой. Подробнее см. *Автоматическое односерверное развертывание FindFace Multi*.
2. Автоматическая установка и настройка FindFace Multi для взаимодействия с удаленными экземплярами `findface-video-worker`. Данный тип установки предназначен для развертывания FindFace Multi в многосерверной среде. Он требует наличия определенных технических навыков и знаний об архитектуре продукта. Чтобы автоматически установить только FindFace Video Worker (`findface-video-worker`) на удаленном сервере, обратитесь к разделу *Развертывание FindFace Video Worker на удаленных серверах*.

3. Полностью настраиваемая установка, которая требует фундаментального понимания архитектуры продукта. См. *Полностью настраиваемая установка*.

Примечание: Если вы выберете тип установки № 3, не забудьте *установить необходимые модели нейронных сетей* вместе с компонентом `findface-extraction-api`.

Вопросы инсталлятора и автоматическое развертывание продукта из файла с параметрами установки

Перед началом активной фазы установки инсталлятор задаст вам несколько вопросов и выполнит ряд автоматических проверок, чтобы убедиться, что сервер соответствует системным требованиям. После ответа на каждый вопрос нужно нажать **Enter**.

Для установки той же конфигурации FindFace Multi на другом сервере используйте автоматическое развертывание из файла с параметрами установки. В этом случае вам не придется снова отвечать на вопросы инсталлятора. Точный путь к файлу с параметрами установки отображается сразу после последнего вопроса инсталлятора, до начала активной фазы установки:

```
[I 2023-02-09 11:13:37,187 main:142] Your answers were saved to /tmp/findface-installer-
↳p01n9sn3.json
```

Важно: Обязательно удалите поля `*.config`, `exp_ip` и `int_ip` перед установкой FindFace Multi на сервере с другим IP-адресом.

Для запуска автоматического развертывания из файла `/tmp/<findface-installer-*>.json`, выполните следующую команду:

```
sudo ./<findface-*>.run -f /tmp/<findface-installer-*>.json
```

Процедуры и первичные инструкции после развертывания

Ознакомьтесь с разделом *Процедуры и первичные инструкции после развертывания*, чтобы узнать, как установить часовой пояс, активировать лицензию и настроить ведение логов. Данный раздел также обучит вас нескольким основным командам, которые помогут вам начать работу с контейнерами FindFace Multi, если вы новичок в Docker.

Важно: Первый после развертывания запуск сервисов с GPU-ускорением `findface-extraction-api` и `findface-video-worker` может занять продолжительное время из-за процесса кэширования (до 45 минут).

Важно: Хотя FindFace Multi предоставляет инструменты для защиты от несанкционированного доступа, они не заменяют правильно настроенный межсетевой экран. Обязательно используйте фаервол, чтобы усилить защиту сети FindFace Multi.

Удаление экземпляра продукта

Для удаления экземпляра FindFace Multi вам потребуется выполнить набор команд. См. раздел *Удаление экземпляра продукта*.

2.3.1 Подготовка сервера Ubuntu

Для того чтобы подготовить сервер на Ubuntu для развертывания FindFace Multi, следуйте приведенным ниже инструкциям, соблюдая последовательность.

Примечание: Для других платформ обратитесь к следующим ресурсам:

- [Драйверы NVIDIA](#)
 - [Docker Engine](#)
 - [Docker Compose](#)
 - [NVIDIA Container Toolkit](#)
-

В этом разделе:

- *GPU: Установка драйверов NVIDIA*
- *Установка продуктов Docker*
- *GPU: Установка NVIDIA Container Runtime*

GPU: Установка драйверов NVIDIA

Первым шагом подготовки сервера является установка драйвера NVIDIA. Данный шаг актуален только для конфигурации GPU. Если ваша конфигурация использует CPU-ускорение, можно переходить к *установке Docker*.

Для работы FindFace Multi с ускорением на GPU потребуется драйвер NVIDIA 530 или более поздней версии. Добавьте репозиторий NVIDIA и установите из него подходящий драйвер.

Предупреждение: Не рекомендуется использовать в качестве альтернативы установщик `.run` с сайта [NVIDIA Driver Downloads](#), так как драйверы, установленные посредством запуска `.run` файла, могут конфликтовать с драйверами, установленными посредством пакетов.

Для установки драйвера 530 из репозитория выполните следующие действия:

1. Установите ключ подписи репозитория:

```
arch=$(uname -m); version=$(. /etc/os-release; echo $ID$VERSION_ID | sed -r 's/\./\//g'
↪'); sudo bash -c \
"sudo apt-key adv --fetch-keys https://developer.download.nvidia.com/compute/cuda/
↪repos/$ID$version/$arch/3bf863cc.pub \
&& apt update"
```

2. Установите aptitude:

```
sudo apt-get install aptitude
```

3. Выполните установку nvidia-driver-530:

```
sudo aptitude install nvidia-driver-530
```

4. Выполните перезагрузку:

```
sudo reboot
```

Установка продуктов Docker

Продукты Docker должны быть установлены как на CPU-, так и на GPU-серверах. Выполните следующие действия:

1. Обновите индекс пакетов apt и установите пакеты, чтобы разрешить apt использовать репозиторий через HTTPS.

```
sudo apt-get update

sudo apt-get install \
  ca-certificates \
  curl \
  gnupg \
  lsb-release
```

Совет: При выполнении группы команд `sudo apt-get install \` возможны следующие ошибки:

```
E: Could not get lock /var/lib/dpkg/lock-frontent - open (11: Resource temporarily
↳ unavailable)
E: Unable to acquire the dpkg frontend lock (/var/lib/dpkg/lock-frontent), is
↳ another process using it?
```

Существуют два способа разрешить их:

1. Принудительно завершите все процессы apt-get, запущенные в данный момент в системе.

```
sudo killall apt apt-get
```

2. Если предыдущая команда не помогла, выполните приведенный ниже набор команд. Если некоторые из подлежащих удалению каталогов не существуют, просто перейдите к следующей команде.

```
sudo rm /var/lib/apt/lists/lock
sudo rm /var/cache/apt/archives/lock
sudo rm /var/lib/dpkg/lock
sudo rm /var/lib/dpkg/lock-frontent
sudo dpkg --configure -a
```

2. Добавьте на сервер официальный ключ GPG для Docker (ключ GNU Privacy Guard).

```
sudo mkdir -m 0755 -p /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /
↳etc/apt/keyrings/docker.gpg
```

3. Настройте репозиторий Docker.

```
echo \
  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg]
↳https://download.docker.com/linux/ubuntu \
  $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/
↳null
```

4. Повторно обновите индекс пакетов apt.

```
sudo apt-get update
```

Совет: Если вы получили ошибку GPG при выполнении данной команды, попробуйте предоставить разрешение на чтение для файла открытого ключа Docker перед обновлением индекса пакетов.

```
sudo chmod a+r /etc/apt/keyrings/docker.gpg
sudo apt-get update
```

5. Установите 24.* версии продуктов Docker.

```
sudo apt-get install docker-ce=5:24* docker-ce-cli=5:24* containerd.io docker-
↳buildx-plugin
```

6. Проверьте успешность установки Docker. Следующая команда загружает тестовый образ и запускает его в контейнере. Когда контейнер запускается, он печатает подтверждающее сообщение и завершает работу.

```
sudo docker run hello-world

Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
2db29710123e: Pull complete
Digest: sha256:aa0cc8055b82dc2509bed2e19b275c8f463506616377219d9642221ab53cf9fe
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.
```

(continues on next page)

(продолжение с предыдущей страницы)

To try something more ambitious, you can run an Ubuntu container with:

```
$ docker run -it ubuntu bash
```

Share images, automate workflows, and more with a free Docker ID:

<https://hub.docker.com/>

For more examples and ideas, visit:

<https://docs.docker.com/get-started/>

7. Установите docker-compose.

```
sudo curl -SL https://github.com/docker/compose/releases/download/v2.15.1/docker-
↪compose-linux-x86_64 -o /usr/local/bin/docker-compose
sudo chmod +x /usr/local/bin/docker-compose
sudo ln -s /usr/local/bin/docker-compose /usr/bin/docker-compose
```

8. Выполните процедуры после установки Docker Engine, чтобы упростить свою будущую работу с контейнерами Docker и *FindFace Multi*. Как только вы будете в состоянии управлять Docker как пользователь без полномочий root, вам больше не понадобится применять `sudo` в командах, связанных с Docker.

```
sudo groupadd docker
sudo usermod -aG docker $USER
newgrp docker
```

9. Внесите изменения в конфигурационный файл сервиса Docker для настройки сети.

```
sudo su
BIP=10.$((RANDOM % 256)).$((RANDOM % 256)).1
cat > /etc/docker/daemon.json <<EOF
{
    "bip": "$BIP/24",
    "fixed-cidr": "$BIP/24"
}
EOF
```

GPU: Установка NVIDIA Container Runtime

Для развертывания контейнеризованного FindFace Multi с ускорением на GPU, потребуется установить NVIDIA Container Runtime. Мы рекомендуем установить NVIDIA Container Toolkit, который включает данную среду выполнения. Выполните следующие действия:

1. Укажите репозиторий и установите из него NVIDIA Container Toolkit, выполнив следующие команды.

```
distribution=$(. /etc/os-release;echo $ID$VERSION_ID) \
    && curl -fsSL https://nvidia.github.io/libnvidia-container/gpgkey | sudo gpg -
↪dearmor -o /usr/share/keyrings/nvidia-container-toolkit-keyring.gpg \
    && curl -s -L https://nvidia.github.io/libnvidia-container/$distribution/
↪libnvidia-container.list | \
    sed 's#deb https://#deb [signed-by=/usr/share/keyrings/nvidia-container-
↪toolkit-keyring.gpg] https://#g' | \
```

(continues on next page)

(продолжение с предыдущей страницы)

```
sudo tee /etc/apt/sources.list.d/nvidia-container-toolkit.list
sudo apt-get update
sudo apt-get install -y nvidia-container-toolkit
sudo nvidia-ctk runtime configure --runtime=docker
sudo systemctl restart docker
```

2. Внесите изменения в конфигурационный файл сервиса Docker для настройки сети и использования среды NVIDIA Container Runtime, установленной вместе с NVIDIA Container Toolkit.

```
sudo su
BIP=10.$((RANDOM % 256)).$((RANDOM % 256))
cat > /etc/docker/daemon.json <<EOF
{
  "default-address-pools":
  [
    {"base": "$BIP.0/16", "size": 24}
  ],
  "bip": "$BIP.1/24",
  "fixed-cidr": "$BIP.0/24",
  "runtimes": {
    "nvidia": {
      "path": "nvidia-container-runtime",
      "runtimeArgs": []
    }
  },
  "default-runtime": "nvidia"
}
EOF
```

3. Перезапустите Docker.

```
systemctl restart docker
```

Теперь все готово для установки FindFace Multi. См. следующие разделы:

- *Автоматическое односерверное развертывание FindFace Multi*
- *Развертывание FindFace Video Worker на удаленных серверах*
- *Полностью настраиваемая установка*

2.3.2 Подготовка сервера CentOS 7

Для того чтобы подготовить сервер на CentOS 7 для развертывания FindFace Multi, следуйте приведенным ниже инструкциям, соблюдая последовательность.

Примечание: Для других платформ обратитесь к следующим ресурсам:

- [Драйверы NVIDIA](#)
- [Docker Engine](#)
- [Docker Compose](#)
- [NVIDIA Container Toolkit](#)

В этом разделе:

- *Установка обновлений*
- *GPU: Установка драйверов NVIDIA*
- *Установка продуктов Docker*
- *GPU: Установка NVIDIA Container Runtime*

Установка обновлений

1. Запустите обновление системы и перезагрузите сервер.

```
sudo yum update
sudo reboot
```

2. Установите fuse с помощью следующей команды.

```
sudo yum -y install fuse
```

GPU: Установка драйверов NVIDIA

Первым шагом подготовки сервера является установка драйвера NVIDIA. Данный шаг актуален только для конфигурации GPU. Если ваша конфигурация использует CPU-ускорение, можно переходить к *установке Docker*.

Для работы FindFace Multi с ускорением на GPU потребуется драйвер NVIDIA 530 или более поздней версии. Загрузите соответствующий установщик `.run` с сайта [NVIDIA Driver Downloads](https://www.nvidia.com/en-us/drivers/anonymous/).

Так как установка производится через установщик `.run`, необходимо установить следующие зависимости:

```
sudo yum install kernel-devel gcc kernel-headers
```

Установка продуктов Docker

Продукты Docker должны быть установлены как на CPU-, так и на GPU-серверах. Выполните следующие действия:

1. Установите пакет `yum-utils` (который предоставляет утилиту `yum-config-manager`) и настройте репозиторий.

```
sudo yum install -y yum-utils
sudo yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-
↪ce.repo
```

2. Установите 24.* версии продуктов Docker.

```
sudo yum install docker-ce-3:24* docker-ce-cli-1:24* docker-ce-rootless-extras-24*  
↪containerd.io docker-buildx-plugin docker-compose-plugin
```

3. Перезапустите Docker.

```
sudo systemctl start docker  
sudo systemctl enable docker
```

4. Проверьте успешность установки Docker. Следующая команда загружает тестовый образ и запускает его в контейнере. Когда контейнер запускается, он печатает подтверждающее сообщение и завершает работу.

```
sudo docker run hello-world
```

Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
2db29710123e: Pull complete
Digest: sha256:aa0cc8055b82dc2509bed2e19b275c8f463506616377219d9642221ab53cf9fe
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:

1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
(amd64)
3. The Docker daemon created a new container from that image which runs the executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
\$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
<https://hub.docker.com/>

For more examples and ideas, visit:
<https://docs.docker.com/get-started/>

5. Установите docker-compose.

```
sudo curl -SL https://github.com/docker/compose/releases/download/v2.15.1/docker-  
↪compose-linux-x86_64 -o /usr/local/bin/docker-compose  
sudo chmod +x /usr/local/bin/docker-compose  
sudo ln -s /usr/local/bin/docker-compose /usr/bin/docker-compose
```

6. Выполните [процедуры после установки Docker Engine](#) , чтобы упростить свою будущую работу с контейнерами Docker и *FindFace Multi*. Как только вы будете в состоянии управлять Docker как пользователь без полномочий root, вам больше не понадобится применять `sudo` в командах, связанных с Docker.

```
sudo groupadd docker
sudo usermod -aG docker $USER
newgrp docker
```

- Внесите изменения в конфигурационный файл сервиса Docker для настройки сети и использования `devicemapper`.

```
sudo su
BIP=10.$((RANDOM % 256)).$((RANDOM % 256)).1
cat > /etc/docker/daemon.json <<EOF
{
    "bip": "$BIP/24",
    "fixed-cidr": "$BIP/24",
    "storage-driver": "devicemapper"
}
EOF
```

GPU: Установка NVIDIA Container Runtime

Для развертывания контейнеризованного FindFace Multi с ускорением на GPU, потребуется установить NVIDIA Container Runtime. Мы рекомендуем установить NVIDIA Container Toolkit, который включает данную среду выполнения. Выполните следующие действия:

- Укажите репозиторий и установите из него NVIDIA Container Toolkit, выполнив следующие команды.

```
distribution=$(. /etc/os-release;echo $ID$VERSION_ID) \
    && curl -s -L https://nvidia.github.io/libnvidia-container/$distribution/
    ↪libnvidia-container.repo | sudo tee /etc/yum.repos.d/nvidia-container-toolkit.repo
sudo yum clean expire-cache
sudo yum install -y nvidia-container-toolkit
sudo nvidia-ctl runtime configure --runtime=docker
sudo systemctl restart docker
```

- Внесите изменения в конфигурационный файл сервиса Docker для настройки сети, использования `devicemapper` и среды NVIDIA Container Runtime, установленной вместе с NVIDIA Container Toolkit.

```
sudo su
BIP=10.$((RANDOM % 256)).$((RANDOM % 256))
cat > /etc/docker/daemon.json <<EOF
{
    "default-address-pools":
        [
            {"base": "$BIP.0/16", "size": 24}
        ],
    "bip": "$BIP.1/24",
    "fixed-cidr": "$BIP.0/24",
    "runtimes": {
        "nvidia": {
            "path": "nvidia-container-runtime",
            "runtimeArgs": []
        }
    }
}
```

(continues on next page)

(продолжение с предыдущей страницы)

```
    }  
  },  
  "default-runtime": "nvidia",  
  "storage-driver": "devicemapper"  
}  
EOF
```

3. Перезапустите Docker.

```
systemctl restart docker
```

Теперь все готово для установки FindFace Multi. См. следующие разделы:

- *Автоматическое односерверное развертывание FindFace Multi*
- *Развертывание FindFace Video Worker на удаленных серверах*
- *Полностью настраиваемая установка*

2.3.3 Подготовка сервера Debian 11

Для того чтобы подготовить сервер на Debian 11 для развертывания FindFace Multi, следуйте приведенным ниже инструкциям, соблюдая последовательность.

Примечание: Для других платформ обратитесь к следующим ресурсам:

- [Драйверы NVIDIA](#)
 - [Docker Engine](#)
 - [Docker Compose](#)
 - [NVIDIA Container Toolkit](#)
-

В этом разделе:

- *Установка FUSE*
- *GPU: Установка драйверов NVIDIA*
- *Установка продуктов Docker*
- *GPU: Установка NVIDIA Container Runtime*

Установка FUSE

1. Установите файловую систему в пользовательском окружении (FUSE) с помощью следующей команды.

```
sudo apt install fuse -y
```

GPU: Установка драйверов NVIDIA

Первым шагом подготовки сервера является установка драйвера NVIDIA. Данный шаг актуален только для конфигурации GPU. Если ваша конфигурация использует CPU-ускорение, можно переходить к [установке Docker](#).

Для работы FindFace Multi с ускорением на GPU потребуется драйвер NVIDIA 530 или более поздней версии. Загрузите соответствующий установщик `.run` с сайта [NVIDIA Driver Downloads](#).

Так как установка производится через установщик `.run`, необходимо установить следующие зависимости:

```
sudo apt install linux-headers-$(uname -r)
sudo apt install build-essential
```

Установка продуктов Docker

Продукты Docker должны быть установлены как на CPU-, так и на GPU-серверах. Выполните следующие действия:

1. Обновите индекс пакетов `apt` и установите пакеты, чтобы разрешить `apt` использовать репозиторий через HTTPS.

```
sudo apt-get update
sudo apt-get install ca-certificates curl gnupg
```

2. Добавьте на сервер официальный ключ GPG для Docker (ключ GNU Privacy Guard).

```
sudo install -m 0755 -d /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/debian/gpg | sudo gpg --dearmor -o /
↳etc/apt/keyrings/docker.gpg
sudo chmod a+r /etc/apt/keyrings/docker.gpg
```

3. Настройте репозиторий Docker.

```
echo \
  "deb [arch="$(dpkg --print-architecture)" signed-by=/etc/apt/keyrings/docker.gpg]
↳https://download.docker.com/linux/debian \
  "$(. /etc/os-release && echo "$VERSION_CODENAME")" stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

4. Повторно обновите индекс пакетов `apt`.

```
sudo apt-get update
```

5. Установите 24.* версии продуктов Docker.

```
sudo apt-get install docker-ce=5:24* docker-ce-cli=5:24* containerd.io docker-  
↳ buildx-plugin
```

6. Проверьте успешность установки Docker.

```
sudo docker run hello-world
```

Эта команда загружает тестовый образ и запускает его в контейнере. Когда контейнер запускается, он печатает подтверждающее сообщение и завершает работу.

7. Установите docker-compose.

```
sudo curl -SL https://github.com/docker/compose/releases/download/v2.15.1/docker-  
↳ compose-linux-x86_64 -o /usr/local/bin/docker-compose  
sudo chmod +x /usr/local/bin/docker-compose  
sudo ln -s /usr/local/bin/docker-compose /usr/bin/docker-compose
```

8. Выполните процедуры после установки Docker Engine, чтобы упростить свою будущую работу с контейнерами Docker и *FindFace Multi*. Как только вы будете в состоянии управлять Docker как пользователь без полномочий root, вам больше не понадобится применять `sudo` в командах, связанных с Docker.

```
sudo groupadd docker  
sudo usermod -aG docker $USER  
newgrp docker
```

9. Внесите изменения в конфигурационный файл сервиса Docker для настройки сети.

```
sudo su  
BIP=10.$((RANDOM % 256)).$((RANDOM % 256)).1  
cat > /etc/docker/daemon.json <<EOF  
{  
  "bip": "$BIP/24",  
  "fixed-cidr": "$BIP/24"  
}  
EOF
```

GPU: Установка NVIDIA Container Runtime

Для развертывания контейнеризованного FindFace Multi с ускорением на GPU, потребуется установить NVIDIA Container Runtime. Мы рекомендуем установить NVIDIA Container Toolkit, который включает данную среду выполнения. Выполните следующие действия:

1. Укажите репозиторий и установите из него NVIDIA Container Toolkit, выполнив следующие команды.

```
distribution=$(. /etc/os-release;echo $ID$VERSION_ID) \  
  && curl -fsSL https://nvidia.github.io/libnvidia-container/gpgkey | sudo gpg -  
↳ -dearmor -o /usr/share/keyrings/nvidia-container-toolkit-keyring.gpg \  
  && curl -s -L https://nvidia.github.io/libnvidia-container/$distribution/  
↳ libnvidia-container.list | \  
    sed 's#deb https://#deb [signed-by=/usr/share/keyrings/nvidia-container-  
↳ toolkit-keyring.gpg] https://#g' | \  
    sudo tee /etc/apt/sources.list.d/nvidia-container-toolkit.list
```

(continues on next page)

(продолжение с предыдущей страницы)

```

sudo tee /etc/apt/sources.list.d/nvidia-container-toolkit.list
sudo apt-get update
sudo apt-get install -y nvidia-container-toolkit
sudo nvidia-ctk runtime configure --runtime=docker
sudo systemctl restart docker

```

2. Переключитесь на учетную запись суперпользователя.

```
sudo su
```

3. Внесите изменения в конфигурационный файл сервиса Docker для настройки сети и использования среды NVIDIA Container Runtime, установленной вместе с NVIDIA Container Toolkit.

```

BIP=10.$((RANDOM % 256)).$((RANDOM % 256))
cat > /etc/docker/daemon.json <<EOF
{
  "default-address-pools":
  [
    {"base": "$BIP.0/16", "size": 24}
  ],
  "bip": "$BIP.1/24",
  "fixed-cidr": "$BIP.0/24",
  "runtimes": {
    "nvidia": {
      "path": "nvidia-container-runtime",
      "runtimeArgs": []
    }
  },
  "default-runtime": "nvidia"
}
EOF

```

4. Перезапустите Docker.

```
systemctl restart docker
```

Теперь все готово для установки FindFace Multi. См. следующие разделы:

- *Автоматическое односерверное развертывание FindFace Multi*
- *Развертывание FindFace Video Worker на удаленных серверах*
- *Полностью настраиваемая установка*

2.3.4 Автоматическое односерверное развертывание FindFace Multi

Для автоматического односерверного развертывания FindFace Multi следуйте приведенным ниже инструкциям. Будучи самым простым, данный тип установки отлично подходит для начала работы с FindFace Multi. Перед процедурой обязательно учтите *системные требования* и *подготовьте сервер*.

Важно: Для успешного функционирования системы после установки из инсталлятора IP-адрес сервера должен быть статическим. Для того чтобы сделать IP-адрес статическим, откройте файл `/etc/network/interfaces` и измените текущую запись для основного сетевого интерфейса так, как показано в примере ниже. Не забудьте заменить адреса в примере на актуальные с учетом настроек сети.

```
sudo vi /etc/network/interfaces

iface eth0 inet static
address 192.168.112.144
netmask 255.255.255.0
gateway 192.168.112.254
dns-nameservers 192.168.112.254
```

Перезапустите сетевые интерфейсы.

```
sudo service networking restart
```

С осторожностью редактируйте файл `etc/network/interfaces`. Перед тем как приступить к редактированию, ознакомьтесь с [инструкцией по настройке сетей Ubuntu](#).

Выполните следующие действия:

1. Загрузите файл инсталлятора `findface-*.run`.
2. Поместите файл `.run` в любой каталог на сервере установки (например, `/home/username`).
3. Из данного каталога сделайте файл `.run` исполняемым.

Примечание: Вместо `findface-*` укажите актуальное имя файла.

```
chmod +x findface-*.run
```

4. Запустите файл `.run`.

```
sudo ./findface-*.run
```

Инсталлятор задаст вам несколько вопросов, после чего проверит, соответствует ли сервер системным требованиям. Ответьте на вопросы надлежащим образом, после ответа каждый раз нажимая `Enter`. Вопросы и ответы следующие:

1. B: Which product should be installed? (Какой продукт следует установить?)

O: 1

```
1. [multi  ] FindFace Multi
2. [server ] FindFace Server
3. [video-worker] FindFace Video Worker

(default: multi)
product> 1
```

2. B: Please choose installation type: (Пожалуйста, выберите тип установки:)

O: 1

```
- 1 [stand-alone ] Single Server
- 2 [multi-worker] Single Server, Multiple video workers
- 3 [images      ] Don't install anything, just load the images
- 4 [custom      ] Fully customized installation
```

(continues on next page)

(продолжение с предыдущей страницы)

```
(default: stand-alone)
type> 1
```

3. B: Directory to install into: (Директория установки:)

О: Укажите директорию установки FindFace Multi. По умолчанию установка выполняется в /opt/findface-multi. Нажмите **Enter** для подтверждения. В противном случае укажите нужную директорию и нажмите **Enter**.

```
Directory to install into:
(default: /opt/findface-multi)
dest_dir>
```

4. B: Do you want to install Video Recorder?(y/n) (Хотите ли вы установить Видеомониторинг?)

О: у или n, в зависимости от ваших нужд.

```
Do you want to install Video Recorder?(y/n)
install_video_recorder> n
```

5. B: Found X interface(s). Which one should we announce as our external address? (Было найдено X сетевых интерфейсов. Какой интерфейс следует использовать в качестве внешнего IP-адреса?)

О: Выберите интерфейс, который вы собираетесь использовать в качестве IP-адреса FindFace.

```
Found 1 interface(s). Which one should we announce as our external address?

- 1 [lo      ] 127.0.0.1
- 2 [ens3    ] 192.168.112.254

(default: 192.168.112.254)
ext_ip.advertised> 2
```

6. B: Which variant of Video Worker should be installed? (Какой вариант Video Worker следует установить?)

О: Укажите тип пакета findface-video-worker, CPU или GPU.

```
Which variant of Video Worker should be installed?

- 1 [cpu] CPU-based implementation, slower but doesn't require GPU
- 2 [gpu] CUDA-based implementation of video detector, requires NVIDIA GPU

(default: cpu)
findface-video-worker.variant> 1
```

7. B: Which variant of Extraction API should be installed? (Какой вариант Extraction API следует установить?)

О: Укажите тип пакета findface-extraction-api, CPU или GPU.

```
Which variant of Extraction API should be installed?
```

- 1 [cpu] CPU-only implementation, slower but doesn't require GPU
- 2 [gpu] CUDA-based implementation, faster, requires NVIDIA GPU (supports ↵
↵ both CPU and GPU models)

```
(default: cpu)
```

```
findface-extraction-api.variant> 1
```

8. B: Do you want to configure detectors and features right now?(y/n) (Хотите ли вы настроить детекторы и распознавание атрибутов объектов сейчас?)

О: Мы рекомендуем установить и настроить функциональность обнаружения объектов и распознавания атрибутов объектов на этапе установки. Ответьте **y**, чтобы начать процесс. Вы можете пропустить данный шаг, ответив **n** и выполнив необходимые действия позже, следуя инструкциям в разделах *Распознавание лиц и их атрибутов*, *Распознавание транспортных средств и их атрибутов* и *Распознавание силуэтов и их атрибутов*.

```
Do you want to configure detectors and features right now?(y/n)
```

```
configure> y
```

9. B: Please select detectors to install: (Пожалуйста, выберите детекторы для установки:)

О: Данный вопрос появляется после запроса на установку и настройку функций обнаружения объектов и распознавания атрибутов объектов. В форме множественного выбора детектор лиц отмечен по умолчанию. Введите соответствующий номер, чтобы выбрать невыбранный детектор и наоборот. Например, чтобы добавить в конфигурацию детектор силуэтов и детектор транспортных средств, введите 2 3. Введите **done**, чтобы продолжить.

```
Please select detectors to install:
```

- 1 [v] Face
- 2 [] Body
- 3 [] Car

```
Enter keyword to select matching choices or -keyword to clear selection.
```

```
Enter "done" to save your selection and proceed to another step.
```

```
detectors>2 3
```

- 1 [v] Face
- 2 [v] Body
- 3 [v] Car

```
Enter keyword to select matching choices or -keyword to clear selection.
```

```
Enter "done" to save your selection and proceed to another step.
```

```
detectors> done
```

10. B: Enable liveness and attempt to continue installation?(y/n) (Включить детектор витальности лиц liveness и продолжить установку?)

О: Данный вопрос появляется после запроса на установку и настройку функций обнаружения объектов и распознавания атрибутов объектов. Чтобы установить встроенный детектор витальности лиц, введите **y**. В противном случае введите **n**.

```
Enable liveness and attempt to continue installation?(y/n)
enable_liveness> y
```

11. В: Please select face features to install: (Пожалуйста, выберите атрибуты лиц для установки:)

О: Данный вопрос появляется после запроса на установку и настройку функций обнаружения объектов и распознавания атрибутов объектов. По умолчанию установке подлежат все атрибуты лиц. Ответьте **done** для подтверждения. Если определенный атрибут не нужен, вы можете ввести ключевое слово (порядковый номер), связанное с ним. Например, введите 7, чтобы исключить распознавание положения головы. Затем введите **done**.

Вопросы по распознаванию атрибутов силуэтов и транспортных средств будут аналогичны, с несколькими вариантами ответов в зависимости от запрашиваемого типа детектора.

```
Please select face features to install:
```

- 1 [v] Age
- 2 [v] Gender
- 3 [v] Emotions
- 4 [v] Beard
- 5 [v] Glasses
- 6 [v] Medicine masks
- 7 [v] Headpose

```
Enter keyword to select matching choices or -keyword to clear selection.
```

```
Enter "done" to save your selection and proceed to another step.
```

```
face_features> done
```

Вопросы по распознаванию атрибутов силуэтов и транспортных средств будут аналогичны, с несколькими вариантами ответов в зависимости от запрашиваемого типа детектора.

12. В: Please set findface-multi admin password (Пожалуйста, установите пароль для администратора FindFace Multi)

О: Задайте пароль для Супер Администратора (**superuser**).

```
Please set findface-multi admin password
findface-multi-admin-password> admin
```

Инсталлятор извлечет образы FindFace Multi из реестра Ntechlab и запустит следующие сервисы в контейнерах Docker:

Сер-вис	Контей-нер	Конфигурация
findface-multi-pause	findface-multi-findface-multi-pause-1	Запускается
nats-jetstream	findface-multi-nats-jetstream-1	Запускается
mongodb	findface-multi-mongodb-1	Устанавливается вместе с Видеомagneитофоном. Запускается
findface-ntls	findface-multi-findface-ntls-1	Запускается
nats	findface-multi-nats-1	Запускается
postgresql	findface-multi-postgresql-1	Запускается
memcached	findface-multi-memcached-1	Запускается
findface-upload	findface-multi-findface-upload-1	Запускается
etcd	findface-multi-etcd-1	Запускается
findface-sf-api	findface-multi-findface-sf-api-1	Запускается
findface-extraction-api	findface-multi-findface-extraction-api-1	Запускается (CPU/GPU-ускорение).
findface-tarantool-server-shard-*	findface-multi-findface-tarantool-server-shard-1	Запускается. Количество экземпляров (шардов) рассчитывается по формуле: $N = \min(\max(\min(\text{mem_mb} // 2000, \text{cpu_cores}), 1), 16 * \text{cpu_cores})$. Т. е. количество равно размеру оперативной памяти в Мб, разделенному на 2000, или количеству физических ядер процессора (но не менее 1 шарда), или же количеству физических ядер процессора, умноженному на 16, если первое полученное значение больше.
findface-video-manager	findface-multi-findface-video-manager	Запускается

По завершении установки в консоль будет выведена информация, необходимая для использования FindFace:

Совет: Обязательно сохраните эти данные: они вам понадобятся.

```
#####
#                               Installation is complete                               #
#####
- all configuration and data is stored in /opt/findface-multi
- upload your license to http://192.168.0.90/#/license/
- user interface: http://192.168.0.90/
superuser:      admin
documentation:  http://192.168.0.90/doc/
Installation logfile: /tmp/installer_run_2570703278.log
```

5. Выполните *процедуры после развертывания*.

Совет: Для установки той же конфигурации FindFace Multi на другом сервере используйте автоматическое развертывание из файла с параметрами установки. В этом случае вам не придется снова отвечать на вопросы инсталлятора. Точный путь к файлу с параметрами установки отображается сразу после последнего вопроса инсталлятора, до начала активной фазы установки:

```
[I 2023-02-09 11:13:37,187 main:142] Your answers were saved to /tmp/findface-installer-
↳ p01n9sn3.json
```

Обязательно удалите поля *.config, exp_ip и int_ip перед установкой FindFace Multi на сервере с другим IP-адресом.

Для запуска автоматического развертывания из файла /tmp/<findface-installer-*>.json, выполните следующую команду:

```
sudo ./<findface-*>.run -f /tmp/<findface-installer-*>.json
```

2.3.5 Развертывание FindFace Video Worker на удаленных серверах

Вы можете установить экземпляры findface-video-worker отдельно от основного сервера FindFace Multi при создании многосерверной среды.

Важно: Прежде чем приступить к установке сервиса findface-video-worker на удаленных серверах, выполните следующие действия:

1. Разрешите доступ к лицензионному серверу findface-ntls с любого IP-адреса. Для этого откройте файл конфигурации /opt/findface-multi/configs/findface-ntls/findface-ntls.yaml на сервере с findface-ntls и установите listen = 0.0.0.0:3133. Перезапустите контейнер findface-multi-findface-ntls-1.

```
sudo vi /opt/findface-multi/configs/findface-ntls/findface-ntls.yaml

listen: 0.0.0.0:3133
```

```
sudo docker container restart findface-multi-findface-ntls-1
```

2. Разрешите доступ к сервису `findface-video-manager` с любого IP-адреса. Для этого откройте файл конфигурации `/opt/findface-multi/configs/findface-video-manager/findface-video-manager.yaml` на сервере с `findface-video-manager` и установите `listen: 0.0.0.0:18810` и `rpc:listen: 0.0.0.0:18811`. Перезапустите контейнер `findface-multi-findface-video-manager-1`.

```
sudo vi /opt/findface-multi/configs/findface-video-manager/findface-video-manager.  
→yaml  
  
listen: 0.0.0.0:18810  
...  
rpc:  
  listen: 0.0.0.0:18811
```

```
sudo docker container restart findface-multi-findface-video-manager-1
```

3. На сервере FindFace Multi откройте файл конфигурации `/opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py` и убедитесь, что параметр `ROUTER_URL` содержит внешний IP-адрес сервера FindFace Multi, а не `localhost`. Экземпляры `findface-video-worker` на удаленных серверах будут использовать данный адрес для отправки объектов.

```
sudo vi /opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py  
  
...  
  
'ROUTER_URL': 'http://192.168.0.12',  
  
...
```

Для отдельной установки сервиса `findface-video-worker` выполните следующие действия:

Совет: Перед тем как приступить к разворачиванию, обязательно ознакомьтесь с *системными требованиями*, а также *подготовьте сервер*.

1. Загрузите файл инсталлятора `findface-*.run`.
2. Поместите файл `.run` в любой каталог на сервере установки (например, `/home/username`).
3. Из данного каталога сделайте файл `.run` исполняемым.

Примечание: Вместо `findface-*` укажите актуальное имя файла.

```
chmod +x findface-*.run
```

4. Запустите файл `.run`.

```
sudo ./findface-*.run
```

Инсталлятор задаст вам несколько вопросов, после чего проверит, соответствует ли сервер системным требованиям. Ответьте на вопросы надлежащим образом, после ответа каждый раз нажимая **Enter**. Вопросы и ответы следующие:

1. B: Which product should be installed? (Какой продукт следует установить?)

O: 3

```
1. [multi  ] FindFace Multi
2. [server ] FindFace Server
3. [video-worker] FindFace Video Worker

(default: multi)
product> 3
```

2. B: Which variant of Video Worker should be installed? (Какой вариант Video Worker следует установить?)

O: Укажите тип пакета findface-video-worker, CPU или GPU.

```
Which variant of Video Worker should be installed?

- 1 [cpu] CPU-based implementation, slower but doesn't require GPU
- 2 [gpu] CUDA-based implementation of video detector, requires NVIDIA GPU
```

3. B: Found X interface(s). Which one should we announce as our inter-service communication address? (Было найдено X сетевых интерфейсов. Какой интерфейс следует использовать в качестве адреса для межсервисной коммуникации?)

O: Выберите интерфейс по умолчанию 2 для подключения к серверу FindFace Multi. Мы не рекомендуем использовать localhost.

```
Found 1 interface(s). Which one should we announce as our external address?

- 1 [lo      ] 127.0.0.1
- 2 [ens3    ] 192.168.112.254

(default: 192.168.112.254)
inter_ip.advertised> 2
```

4. B: Please enter FF.Multi or FF.Server IP address: (Пожалуйста, введите адрес сервера FindFace Multi или FindFace Enterprise Server)

O: Укажите IP-адрес сервера FindFace Multi.

```
Please enter FF.Multi or FF.Server IP address:
server_addr> 192.168.112.25
```

После этого процесс установки будет автоматически запущен.

Важно: Если вы решили установить findface-ntls и/или findface-video-manager на серверах, отличных от серверов с findface-multi-legacy, укажите их IP-адреса в файле конфигурации /opt/findface-multi/configs/findface-video-worker/findface-video-worker.yaml после установки.

```
/opt/findface-multi/configs/findface-video-worker/findface-video-worker.yaml
```

В параметре ntls-addr укажите IP-адрес сервера findface-ntls.

```
ntls_addr: 127.0.0.1:3133
```

В параметре `mgr-static` укажите IP-адрес сервера `findface-video-manager`, который будет обеспечивать `findface-video-worker` настройками и списком видеопотоков для обработки.

```
mgr:
  static: 127.0.0.1:18811
```

Совет: Чтобы установить экземпляр `findface-video-worker` на другом сервере, используйте автоматическое развертывание из файла с параметрами установки. В этом случае вам не придется повторно отвечать на вопросы инсталлятора. Точный путь к файлу с параметрами установки отображается сразу после последнего вопроса инсталлятора, перед началом активной фазы установки:

```
[I 2023-02-09 11:13:37,187 main:142] Your answers were saved to /tmp/findface-installer-
↳ p01n9sn3.json
```

Для запуска автоматического развертывания из файла `/tmp/<findface-installer-*>.json`, выполните следующую команду:

```
sudo ./<findface-*>.run -f /tmp/<findface-installer-*>.json
```

2.3.6 Пошаговое развертывание Видеомагнитофона

Существуют следующие способы развертывания Видеомагнитофона:

- Автоматически при развертывании FindFace Multi из консольного инсталлятора. Подробнее см. в *Автоматическое односерверное развертывание FindFace Multi*.
- Пошагово. Рекомендуется в многосерверной среде.

Данный раздел посвящен пошаговому развертыванию Видеомагнитофона. Выполните приведенные ниже инструкции, придерживаясь заданного порядка.

Совет: Сначала обязательно ознакомьтесь с *архитектурой* FindFace Multi.

В этом разделе:

- Установка apt-репозитория
- Установка основных компонентов

Установка apt-репозитория

Прежде всего установите apt-репозиторий FindFace следующим образом:

1. Загрузите файл инсталлятора `findface-*.run`.
2. Поместите файл `.run` в любой каталог на сервере установки (например, `/home/username`).
3. Из данного каталога сделайте файл `.run` исполняемым.

```
chmod +x findface-*.run
```

4. Запустите файл `.run`.

```
sudo ./findface-*.run
```

Инсталлятор задаст вам несколько вопросов, после чего проверит, соответствует ли сервер системным требованиям. Ответьте на вопросы надлежащим образом, после ответа каждый раз нажимая **Enter**. Вопросы и ответы следующие:

1. B: Which product should be installed? (Какой продукт следует установить?)

O: 1

```
Which product should be installed?

1. [multi  ] FindFace Multi
2. [server ] FindFace Server
3. [video-worker] FindFace Video Worker

(default: multi)
product> 1
```

2. B: Please choose installation type: (Пожалуйста, выберите тип установки:)

O: 3

```
- 1 [stand-alone ] Single Server
- 2 [multi-worker] Single Server, Multiple video workers
- 3 [images      ] Don't configure or start anything, just load
  ↳ the images and copy the models
- 4 [custom      ] Fully customized installation

(default: stand-alone)
type> 3
```

3. B: Directory to install into: (Директория установки:)

O: Укажите директорию установки FindFace Multi. По умолчанию установка выполняется в `/opt/findface-multi`. Нажмите **Enter** для подтверждения. В противном случае укажите нужную директорию и нажмите **Enter**.

```
Directory to install into:
(default: /opt/findface-multi)
dest_dir>
```

4. B: Select models to install. Note that you will need to accordingly edit extraction-api and tntapi configuration files. At least one of recognition models has to be enabled. (Выберите модели для установки. Обратите внимание, что вам нужно будет отредактировать файлы конфигурации extraction-api и tntapi. Должна быть включена хотя бы одна из моделей распознавания.)

О: По умолчанию все модели нейронных сетей подлежат установке. Вы можете оставить все выбранные модели, введя `done`, или выбрать конкретные модели. Для этого отмените выбор всех моделей из списка, введя `-*` в командной строке, затем выберите необходимые модели, введя их порядковые номера (keyword): например, `1 3 4`. Введите `done`, чтобы сохранить выбранный вами вариант и перейти к следующему шагу. Если FindFace Multi был развернут ранее, модели устанавливать не нужно.

```
Select models to install.
Note that you will need to accordingly edit extraction-api and
↳tntapi configuration files.
At least one of recognition models has to be enabled.

- 1 [ ] ./models/carattr/carattr.categories.v0.cpu.fnk
...
- 91 [ ] ./models/pedrec/pedrec.clio.gpu.fnk

Enter keyword to select matching choices or -keyword to clear
↳selection.
Enter "done" to save your selection and proceed to another step.
findface-data.models> done
```

После этого apt-репозиторий FindFace будет автоматически установлен.

Установка основных компонентов

Для того чтобы установить компоненты Видеомегнитофона, выполните следующие действия:

1. Откройте файл конфигурации `/opt/findface-multi/docker-compose.yaml`. Добавьте `findface-video-storage` и `findface-video-streamer` сервисы.

```
sudo vi /opt/findface-multi/docker-compose.yaml

...
findface-video-storage:
  command: [--config=/etc/findface-video-storage.conf]
  depends_on: [mongodb]
  image: docker.int.ntl/ntech/universe/video-storage:ffserver-8.221216
  network_mode: service:pause
  restart: always
  volumes: ['./configs/findface-video-storage/findface-video-storage.yaml:/etc/
↳findface-video-storage.conf:ro']
findface-video-streamer:
  command: [--config=/etc/findface-video-streamer-cpu.ini]
  depends_on: [findface-ntls, mongodb]
  image: docker.int.ntl/ntech/universe/video-streamer-cpu:ffserver-8.221216
  network_mode: service:pause
```

(continues on next page)

(продолжение с предыдущей страницы)

```
restart: always
volumes: ['./configs/findface-video-streamer/findface-video-streamer.yaml:/etc/
↪findface-video-streamer-cpu.ini:ro',
          './cache/findface-video-streamer:/var/cache/findface/video-streamer']
```

2. Создайте директории.

```
sudo mkdir -p /opt/findface-multi/configs/findface-video-storage/ /opt/findface-
↪multi/configs/findface-video-streamer/
```

3. Создайте файл конфигурации /opt/findface-multi/configs/findface-video-storage/findface-video-storage.yaml и добавьте в него следующее:

```
sudo vi /opt/findface-multi/configs/findface-video-storage/findface-video-storage.
↪yaml

listen: :18611
debug: false
external-address: http://ip_address:18611/
streamer:
  endpoints:
    - 127.0.0.1:9000
chunk-storage:
  type: webdav
webdav:
  timeouts:
    connect: 5s
    response_header: 30s
    overall: 35s
    idle_connection: 10s
  max-idle-conns-per-host: 20
  keepalive: 24h0m0s
  trace: false
  upload-url: http://127.0.0.1:3333/uploads/video_storage
s3:
  endpoint: ''
  bucket-name: ''
  access-key: ''
  secret-access-key: ''
  secure: true
  region: ''
  public-url: ''
  operation-timeout: 30
localfs:
  directory: ''
meta-storage:
  mongo-uri: mongodb://127.0.0.1
  database: video-storage
timings:
  connect: 3s
```

Примечание: В разделе `external-address` необходимо вставить ip-адрес FindFace Multi.

4. Создайте файл конфигурации `/opt/findface-multi/configs/findface-video-streamer/findface-video-streamer.yaml` и добавьте следующее:

```
sudo vi /opt/findface-multi/configs/findface-video-streamer/findface-video-streamer.
↪yaml

streamer:
  port: 9000
  max_backpressure: 33554432
  io_buffer_size: 524288
video_storage:
  url: http://127.0.0.1:18611
  timeout: 6
cache:
  dir: /var/cache/findface/video-streamer
```

5. Запустите контейнеры FindFace Multi:

```
cd /opt/findface-multi/
docker-compose up -d
```

На этом развертывание Видеоманитфона будет завершено. Для того чтобы настроить Видеоманитфон после развертывания, следуйте [инструкции](#).

2.3.7 Полностью настраиваемая установка

Консольный инсталлятор FindFace Multi предоставляет несколько вариантов установки, в том числе полностью настраиваемый вариант (установку отдельно выбранных пакетов). Данный вариант в основном используется при развертывании FindFace Multi в сильно распределенной среде и требует определенного уровня знаний и опыта.

Для запуска полностью настраиваемой установки выполните следующие действия:

1. Загрузите файл инсталлятора `findface-*.run`.
2. Поместите файл `.run` в любой каталог на сервере установки (например, `/home/username`).
3. Из данного каталога сделайте файл `.run` исполняемым.

Примечание: Вместо `findface-*` укажите актуальное имя файла.

```
chmod +x findface-*.run
```

4. Запустите файл `.run`.

```
sudo ./findface-*.run
```

Инсталлятор задаст вам несколько вопросов, после чего проверит, соответствует ли сервер системным требованиям. Ответьте на вопросы надлежащим образом, после ответа каждый раз нажимая `Enter`. Вопросы и ответы следующие:

1. B: Which product should be installed? (Какой продукт следует установить?)

O: 1

```
Which product should be installed?

1. [multi  ] FindFace Multi
2. [server ] FindFace Server
3. [video-worker] FindFace Video Worker

(default: multi)
product> 1
```

2. B: Please choose installation type: (Пожалуйста, выберите тип установки:)

O: 4

```
Please choose installation type:

- 1 [stand-alone ] Single Server
- 2 [multi-worker] Single Server, Multiple video workers
- 3 [images      ] Don't configure or start anything, just load the images
  and copy the models
- 4 [custom      ] Fully customized installation

(default: stand-alone)
type> 4
```

3. B: Directory to install into: (Директория установки:)

O: Укажите директорию установки FindFace Multi. По умолчанию установка выполняется в /opt/findface-multi. Нажмите Enter для подтверждения. В противном случае укажите нужную директорию и нажмите Enter.

```
Directory to install into:
(default: /opt/findface-multi)
dest_dir>
```

4. B: Please enter path to docker-compose binary: (Пожалуйста, введите путь к бинарному файлу docker-compose)

O: Укажите фактический путь к бинарному файлу **docker-compose**. По умолчанию предлагается /usr/local/bin/docker-compose, и это тот путь, который вы получите при установке **docker-compose** *согласно инструкции*. Нажмите Enter для подтверждения. В противном случае укажите другой путь и нажмите Enter.

```
Please enter path to docker-compose binary
(default: /usr/local/bin/docker-compose)
docker_compose>
```

5. B: Found X interface(s). Which one should we announce as our external address? (Было найдено X сетевых интерфейсов. Какой интерфейс следует использовать в качестве внешнего IP-адреса?)

O: Выберите интерфейс, который вы собираетесь использовать в качестве IP-адреса FindFace.

```
Found 1 interface(s). Which one should we announce as our external address?
```

- 1 [lo] 127.0.0.1
- 2 [ens3] 192.168.112.254

```
(default: 192.168.112.254)
```

```
ext_ip.advertised> 2
```

6. В: Found X interface(s). Which one should we announce as our inter-service communication address? (Было найдено X сетевых интерфейсов. Какой интерфейс следует использовать в качестве адреса для межсервисной коммуникации?)

О: Выберите интерфейс для межсервисного взаимодействия.

```
Found 1 interface(s). Which one should we announce as our inter-service_
↪communication address?
```

- 1 [lo] 127.0.0.1
- 2 [ens3] 192.168.112.254

```
(default: 192.168.112.254)
```

```
inter_ip.advertised> 2
```

7. В: Please select FindFace Multi components to install: (Пожалуйста, выберите компоненты FindFace Multi для установки:)

О: Выберите компоненты FindFace для установки. По умолчанию установке подлежат все компоненты. Вы можете оставить все как есть, введя **done**, или выбрать определенные компоненты. Для того чтобы выбрать нужные, сначала снимите выделение со всех компонентов, введя в командной строке **-***, затем введите порядковые номера нужных компонентов через пробел, например: **1 7 13** и т. д. Введите **done** для сохранения выбора и перехода к следующему шагу.

Предупреждение: Не рекомендуется исключать компонент **pause** из установки, так как оставшиеся компоненты останутся без сетевого идентификатора (namespace), к которому можно подключиться.

Если вы исключаете компонент **pause** намеренно, вам потребуется отредактировать файл `/opt/findface-multi/docker-compose.yaml` и указать имя хоста для каждого сервиса в параметре `network_mode`.

```
Please select FindFace Multi components to install:
```

- 1 [v] findface-data - Face recognition models
- ...
- ...

```
Enter keyword to select matching choices or -keyword to clear selection.
```

```
Enter "done" to save your selection and proceed to another step.
```

```
components> done
```

8. Конкретные вопросы, связанные с выбранными компонентами: тип ускорения, необходимое количество экземпляров компонента, модели нейронных сетей и т. д. Если вы затрудняетесь

с ответом, попробуйте найти его в этой документации или отправьте свой вопрос по адресу support@ntechlab.com.

9. В: Please set findface-multi admin password (Пожалуйста, установите пароль для администратора FindFace Multi)

О: Задайте пароль для Супер Администратора (superuser).

```
Please set findface-multi admin password
findface-multi-admin-password> admin
```

Инсталлятор извлечет образы FindFace Multi из реестра Ntechlab и запустит ассоциированные сервисы в контейнерах Docker.

5. Выполните *процедуры после развертывания*.

Совет: Для установки той же конфигурации FindFace Multi на другом сервере используйте автоматическое развертывание из файла с параметрами установки. В этом случае вам не придется снова отвечать на вопросы инсталлятора. Точный путь к файлу с параметрами установки отображается сразу после последнего вопроса инсталлятора, до начала активной фазы установки:

```
[I 2023-02-09 11:13:37,187 main:142] Your answers were saved to /tmp/findface-installer-
↳ p01n9sn3.json
```

Обязательно удалите поля *.config, exp_ip и int_ip перед установкой FindFace Multi на сервере с другим IP-адресом.

Для запуска автоматического развертывания из файла /tmp/<findface-installer-*>.json, выполните следующую команду:

```
sudo ./<findface-*>.run -f /tmp/<findface-installer-*>.json
```

2.3.8 Стандартное развертывание в многосерверной среде

Данный раздел посвящен развертыванию FindFace Multi в многосерверной среде.

Совет: Если после прочтения данного раздела у вас остались вопросы, не стесняйтесь задать их нашим экспертам по адресу support@ntechlab.com.

Важно: В данном разделе не рассматривается развертывание функционала Видеомэгнитофон. Вы можете найти соответствующую пошаговую инструкцию [здесь](#).

Развертывание FindFace Multi на нескольких серверах может быть необходимо по следующим причинам:

- Необходимость в распределении высокой нагрузки при обработке видео.
- Требуется обработка видеопотоков от группы камер в месте их физического расположения.

Примечание: Актуально для сетей гостиниц, магазинов, при наличии нескольких проходных в одном здании и др.

См.также:

Привязка группы камер к экземпляру findface-video-worker

- Необходимость в распределении высокой нагрузки при извлечении векторов признаков.
- В поиске задействовано большое количество объектов, что требует реализации распределенной базы данных.

Перед тем как приступить к разворачиванию, постройте архитектурную схему с учетом будущей нагрузки системы и выделенных под нее аппаратных ресурсов (см. *Требования*). Наиболее распространенной схемой является следующая:

- Центральный сервер с установленными компонентами `findface-ntls`, `findface-security`, `findface-sf-api`, `findface-video-manager`, `findface-upload`, `findface-video-worker`, `findface-extraction-api`, `findface-tarantool-server`, `pause`, а также сторонним программным обеспечением.
- Несколько дополнительных серверов для обработки видео с установленным компонентом `findface-video-worker`.
- (При необходимости) Несколько дополнительных серверов для извлечения векторов признаков (т. н. серверы экстракции) с установленным компонентом `findface-extraction-api`.
- (При необходимости) Дополнительные серверы базы данных с несколькими шардами Tarantool на каждом.

Инструкции в настоящем разделе приведены для описанной выше наиболее часто встречающейся схемы многосерверного разворачивания. В высоконагруженных системах также может потребоваться распределить обработку API-запросов, т. е. организовать несколько серверов `findface-sf-api` и `findface-video-manager`. Данная процедура требует высокого уровня знаний и дополнительного программирования. Пожалуйста, не стесняйтесь обращаться к нашим экспертам за помощью по этому вопросу (support@ntechlab.com).

Разворачивание FindFace Multi в многосерверной среде состоит из следующих этапов:

- *Разворачивание центрального сервера*
- *Разворачивание серверов обработки видео*
- *Разворачивание серверов экстракции*
- *Распределение нагрузки между серверами экстракции*
- *Разворачивание дополнительных серверов базы данных*
- *Настройка сетевого взаимодействия*

Разворачивание центрального сервера

Для разворачивания центрального сервера FindFace Multi выполните следующие действия:

1. На выделенном физическом сервере *установите* FindFace Multi из инсталлятора следующим образом (до разворачивания FindFace Multi не забудьте *подготовить сервер*):
 - Устанавливаемый продукт: FindFace Multi.
 - Тип установки: `Single server, multiple video workers`. В этом случае FindFace Multi будет установлен в качестве центрального сервера и настроен на взаимодействие с дополнительными удаленными экземплярами `findface-video-worker`.

- Тип ускорения `findface-video-worker` (на центральном сервере): CPU или GPU, в зависимости от конфигурации оборудования.
- Тип ускорения `findface-extraction-api` (на центральном сервере): CPU или GPU, в зависимости от конфигурации оборудования.

По завершении установки в консоль будет выведена информация, необходимая для использования FindFace Multi:

```
#####
#                               Installation is complete                               #
#####
- all configuration and data is stored in /opt/findface-multi
- upload your license to http://172.20.77.17/#/license/
- user interface: http://172.20.77.17/
  superuser:      admin
  documentation:  http://172.20.77.17/doc/
```

2. Загрузите файл лицензии FindFace Multi через основной веб-интерфейс `http://<IP_адрес_сервера>#/license`. Для доступа в веб-интерфейс используйте логин и пароль пользователя `superuser`, выведенные в консоли.

Примечание: IP-адрес сервера в ссылках на веб-интерфейсы FindFace имеет вид `127.0.0.1` или `<IP_адрес_в_сети>`, в зависимости от того, принадлежит ли сервер к сети.

Важно: Не передавайте данные `superuser` (Супер Администратора) третьим лицам. Для администрирования системы создайте назначаемого администратора. Отличие назначаемого администратора от Супер Администратора в том, что последний не может лишиться прав администратора даже при смене роли.

3. Разрешите лицензируемым сервисам обращаться к серверу лицензирования `findface-ntls` с любого IP-адреса. Для этого, откройте файл конфигурации `/opt/findface-multi/configs/findface-ntls/findface-ntls.yaml` и установите `listen: 0.0.0.0:3133`. Перезапустите контейнер `findface-multi-findface-ntls-1`.

```
sudo vi /opt/findface-multi/configs/findface-ntls/findface-ntls.yaml
```

```
listen: 0.0.0.0:3133
```

```
sudo docker container restart findface-multi-findface-ntls-1
```

4. Разрешите доступ к сервису `findface-video-manager` с любого IP-адреса. Для этого откройте файл конфигурации `/opt/findface-multi/configs/findface-video-manager/findface-video-manager.yaml` и установите `listen: 0.0.0.0:18810` и `rpc:listen: 0.0.0.0:18811`. Перезапустите контейнер `findface-multi-findface-video-manager-1`.

```
sudo vi /opt/findface-multi/configs/findface-video-manager/findface-video-manager.
↪yaml
```

```
listen: 0.0.0.0:18810
```

```
...
```

(continues on next page)

(продолжение с предыдущей страницы)

```
rpc:
  listen: 0.0.0.0:18811
```

```
sudo docker container restart findface-multi-findface-video-manager-1
```

Развертывание серверов обработки видео

На дополнительном сервере для обработки видео установите экземпляр **findface-video-worker**, руководствуясь *пошаговыми инструкциями*. Ответьте на вопросы инсталлятора следующим образом:

- Устанавливаемый продукт: FindFace Video Worker.
- Тип ускорения **findface-video-worker**: CPU или GPU, в зависимости от конфигурации оборудования.
- FindFace Multi IP address: IP-адрес центрального сервера.

После этого процесс установки будет автоматически запущен. Ответы на вопросы инсталлятора будут сохранены в файл `/tmp/<findface-installer-*>.json`. Используйте данный файл, чтобы установить FindFace Video Worker на других серверах, не отвечая на вопросы инсталлятора повторно. Для этого запустите инсталлятор командой:

```
sudo ./<findface-*>.run -f /tmp/<findface-installer-*>.json
```

Примечание: Если **findface-ntls** и/или **findface-video-manager** установлены на серверах, отличных от сервера **findface-security**, укажите их IP-адреса после установки в файле конфигурации `/opt/findface-multi/configs/findface-video-worker/findface-video-worker.yaml`.

```
sudo vi /opt/findface-multi/configs/findface-video-worker/findface-video-
↪worker.yaml
```

В параметре **ntls-addr** укажите IP-адрес сервера **findface-ntls**.

```
ntls-addr: 127.0.0.1:3133
```

В параметре **mgr -> static** укажите IP-адрес сервера **findface-video-manager**, который будет обеспечивать **findface-video-worker** настройками и списком видеопотоков для обработки.

```
static: 127.0.0.1:18811
```

Развертывание серверов экстракции

На дополнительном сервере экстракции установите экземпляр `findface-extraction-api` из консольного инсталлятора. Ответьте на вопросы инсталлятора следующим образом:

- Устанавливаемый продукт: FindFace Multi.
- Тип установки: Fully customized installation.
- Устанавливаемые компоненты FindFace Multi: `findface-extraction-api`, `findface-data` и `pause`. Для того чтобы их выбрать, сначала снимите выделение со всех компонентов, введя в командной строке `-*`, затем введите порядковые номера `findface-extraction-api`, `findface-data` и `pause` через пробел. Введите `done` для сохранения выбора и перехода к следующему шагу.

Примечание: Компонент `pause` хранит информацию о сетевых идентификаторах (namespaces) других компонентов. Обязательно установите его.

- Тип ускорения `findface-extraction-api`: CPU или GPU.
- Необходимость в изменении файла конфигурации `/opt/findface-multi/configs/findface-extraction-api/findface-extraction-api.yaml`: укажите IP-адрес сервера `findface-ntls`.
- Модели нейронных сетей для установки: CPU/GPU-модели для биометрии лиц (обязательно) и (опционально) CPU/GPU-модели для распознавания атрибутов лица, транспортных средств и их атрибутов, силуэтов и их атрибутов. Удостоверьтесь, что для каждой модели вы указали правильный тип ускорения: CPU или GPU. Он должен совпадать с типом ускорения `findface-extraction-api`. Обратите внимание, что `findface-extraction-api` на CPU может работать только с CPU-моделями, в то время как `findface-extraction-api` на GPU поддерживает как GPU-, так и CPU-модели.

Совет: Подробнее см. *Обзор моделей нейронных сетей, Распознавание лиц и их атрибутов, Распознавание транспортных средств и их атрибутов, Распознавание силуэтов и их атрибутов*.

- Чтобы переместить основной экземпляр `findface-extraction-api` на другой хост, в файле конфигурации `/opt/findface-multi/configs/findface-sf-api/findface-sf-api.yaml` укажите IP-адрес хоста сервера экстракции и установите `listen: 0.0.0.0:18411`.

```
listen: 0.0.0.0:18411
extraction-api:
  timeouts:
    connect: 5s
    response_header: 30s
    overall: 35s
    idle_connection: 10s
  max-idle-conns-per-host: 20
  keepalive: 24h0m0s
  trace: false
  extraction-api: http://172.20.77.19:18666
```

После этого процесс установки будет автоматически запущен. Ответы на вопросы инсталлятора будут сохранены в файл `/tmp/<findface-installer-*>.json`. Используйте данный файл, чтобы установить `findface-extraction-api` на других серверах, не отвечая на вопросы инсталлятора повторно.

```
sudo ./<findface-*>.run -f /tmp/<findface-installer-*>.json
```

После развертывания серверов экстракции *распределите* между ними нагрузку.

Распределение нагрузки между серверами экстракции

Распределение нагрузки между несколькими серверами экстракции выполняется через балансировщик нагрузки. Приведенная ниже пошаговая инструкция демонстрирует балансировку нагрузки с помощью nginx в режиме round-robin для 3-х экземпляров `findface-extraction-api`, расположенных на различных физических серверах. Один экземпляр установлен на центральном сервере FindFace Multi (172.168.1.9), 2 других на дополнительных удаленных серверах (172.168.1.10, 172.168.1.11). Если в системе присутствует большее количество серверов экстракции, балансировка нагрузки выполняется по аналогии.

Совет: Вы можете использовать любой удобный вам балансировщик нагрузки. Руководство по его использованию ищите в соответствующей справочной документации.

Для балансировки нагрузки между экземплярами `findface-extraction-api` выполните следующие действия:

1. Назначьте т. н. сервер шлюза для балансируемой группы серверов экстракции. Им может стать центральный сервер FindFace Multi (рекомендуется) или любой другой сервер с включенным сервисом `findface-sf-api`.

Важно: Вам нужно будет указать IP-адрес шлюза при настройке *распределенной сети* FindFace Multi.

2. На сервере с включенным сервисом `findface-sf-api` создайте в каталоге `/opt/findface-multi/configs/` папку `nginx`, содержащую файл `extapi.conf`. Убедитесь, что файл `extapi.conf` содержит информацию, как в примере ниже. В директиве `upstream` (`upstream extapibackends`) замените примерные IP-адреса на актуальные IP-адреса серверов экстракции. В директиве `server` в параметре `listen` укажите номер слушающего порта сервера шлюза. Вам потребуется указать данный порт при настройке FindFace Multi *network*.

```
upstream extapibackends {
    server 172.168.1.9:18666; ## ``findface-extraction-api`` on principal server
    server 172.168.1.10:18666; ## 1st additional extraction server
    server 127.168.1.11:18666; ## 2nd additional extraction server
}
server {
    listen 18667;
    server_name extapi;
    client_max_body_size 64m;
    location / {
        proxy_pass http://extapibackends;
        proxy_next_upstream error;
    }
    access_log /var/log/nginx/extapi.access_log;
    error_log /var/log/nginx/extapi.error_log;
}
```

3. Обозначьте сервис Nginx в файле `docker-compose.yaml`. Для этого внесите в файл `docker-compose.yaml` запись о контейнере с образом Nginx:

```
sudo vi /opt/findface-multi/docker-compose.yaml

nginx:
  image: nginx:latest
  ports:
    - 18667:18667
  volumes:
    - ./configs/nginx/extapi.conf:/etc/nginx/conf.d/default.conf:ro
```

4. В конфигурационном файле `findface-sf-api` укажите адрес распределителя:

```
sudo vi /opt/findface-multi/configs/findface-sf-api/findface-sf-api.yaml

listen: 0.0.0.0:18411
...
extraction-api: http://172.168.1.9:18667
```

5. Перезапустите контейнеры.

```
cd /opt/findface-multi/
sudo docker-compose down
sudo docker-compose up -d
```

6. На центральном сервере и каждом из дополнительных серверов экстракции откройте файл конфигурации `/opt/findface-multi/configs/findface-extraction-api/findface-extraction-api.yaml`. Замените адрес `localhost` в параметре `listen` на адрес, который вы указали до этого в директиве `upstream extapibackends` файла конфигурации Nginx (`/opt/findface-multi/configs/nginx/extapi.conf`). В нашем примере адрес 1-го дополнительного сервера экстракции должен быть заменен на следующий:

```
sudo vi /opt/findface-multi/configs/findface-extraction-api/findface-extraction-api.
↪yaml

listen: 172.168.1.10:18666
```

7. Перезапустите контейнер `findface-multi-findface-extraction-api-1` на центральном сервере и каждом дополнительном сервере экстракции.

```
sudo docker container restart findface-multi-findface-extraction-api-1
```

Балансировка нагрузки успешно настроена. Не забудьте указать актуальный IP-адрес и слушающий порт сервера шлюза при настройке *распределенной сети* FindFace Multi.

Развертывание дополнительных серверов базы данных

Компонент `findface-tarantool-server` соединяет базу данных векторов признаков на основе Tarantool и компонент `findface-sf-api`, передавая результаты поиска от базы данных в `findface-sf-api` для дальнейшей обработки.

Чтобы увеличить скорость поиска, можно выделить несколько дополнительных серверов под базу данных векторов признаков и создать множество сегментов (шардов) `findface-tarantool-server` на каждом из них. Параллельное функционирование многочисленных шардов приведет к значительному увеличению производительности, поскольку каждый шард может обрабатывать приблизительно до 10 000 000 векторов признаков.

Чтобы развернуть дополнительные серверы базы данных, выполните следующие действия:

1. Установите компонент `findface-tarantool-server` на первом выделенном под базу данных сервере. Компонент `pause` должен быть уже установлен на сервере. В противном случае установите компонент `pause` вместе с компонентом `findface-tarantool-server`. Ответьте на вопросы инсталлятора следующим образом:
 - Устанавливаемый продукт: FindFace Multi.
 - Тип установки: Fully customized installation.
 - Устанавливаемые компоненты FindFace Multi: `findface-tarantool-server`, `pause`. Для того чтобы их выбрать, сначала снимите выделение со всех компонентов, введя в командной строке `-*`, затем введите порядковый номер `findface-tarantool-server`, `pause` (keyword). Введите `done` для сохранения выбора и перехода к следующему шагу.

После этого процесс установки будет автоматически запущен.

В результате установки шарды `findface-tarantool-server` будут автоматически установлены в количестве $N = \min(\max(\min(\text{mem_mb} // 2000, \text{cpu_cores}), 1), 16 * \text{cpu_cores})$. Т. е. количество равно размеру оперативной памяти в Мб, разделенному на 2000, или количеству физических ядер процессора (но не менее 1 шарда), или же количеству физических ядер процессора, умноженному на 16, если первое полученное значение больше.

2. Используйте созданный файл `/tmp/<findface-installer-*>.json`, чтобы установить `findface-tarantool-server` на других серверах, не отвечая на вопросы инсталлятора повторно. Для этого выполните следующую команду:

```
sudo ./<findface-*>.run -f /tmp/<findface-installer-*>.json
```

3. Обязательно укажите IP-адреса и порты шардов при настройке *распределенной сети* FindFace Multi. Для того чтобы узнать номера портов, на каждом сервере базы данных выполните команду:

```
sudo cat /opt/findface-multi/configs/findface-tarantool-server/*shard-* | grep -E "\.
↪start|(listen =)"
```

Будет возвращен следующий результат:

```
listen = '127.0.0.1:32001',
FindFace.start("127.0.0.1", 8101, {
  listen = '127.0.0.1:32002',
FindFace.start("127.0.0.1", 8102, {
  listen = '127.0.0.1:32003',
FindFace.start("127.0.0.1", 8103, {
  listen = '127.0.0.1:32004',
FindFace.start("127.0.0.1", 8104, {
```

Здесь номера портов: 8101, 8102 и т. д.

4. Разрешите каждому шарду доступ к серверу лицензий `findface-ntls`. Для этого измените файл конфигурации для каждого шарда, как показано в примере ниже. Для каждого шарда обязательно установите `0.0.0.0` в секции `FindFace.start`.

```
sudo vi /opt/findface-multi/configs/findface-tarantool-server/shard-00*

-- host:port to bind, HTTP API
FindFace = require("FindFace")
FindFace.start("0.0.0.0", 8101, {
    license_ntls_server="172.23.218.110:3133",
    replication = replication_master,
    spaces = spaces
})
```

5. Откройте конфигурационный файл `/opt/findface-multi/configs/findface-ntls/findface-ntls.yaml` и установите `listen: 0.0.0.0:3133`. Перезапустите контейнер `findface-multi-findface-ntls-1`.

```
sudo vi /opt/findface-multi/configs/findface-ntls/findface-ntls.yaml

listen: 0.0.0.0:3133
license_dir: /ntech/license
proxy: ''
ui: 0.0.0.0:3185
```

```
sudo docker container restart findface-multi-findface-ntls-1
```

6. Внесите изменения в файл конфигурации `/opt/findface-multi/configs/findface-sf-api/findface-sf-api.yaml`. Установите `listen: 0.0.0.0:18411` и обозначьте шарды. Перезапустите контейнер `findface-multi-findface-sf-api-1`.

```
sudo vi /opt/findface-multi/configs/findface-sf-api/findface-sf-api.yaml

listen: 0.0.0.0:18411
extraction-api:
  timeouts:
    connect: 5s
    response_header: 30s
    overall: 35s
    idle_connection: 10s
  max-idle-conns-per-host: 20
  keepalive: 24h0m0s
  trace: false
  extraction-api: http://127.0.0.1:18666
storage-api:
  timeouts:
    connect: 5s
    response_header: 30s
    overall: 35s
    idle_connection: 10s
  max-idle-conns-per-host: 20
  keepalive: 24h0m0s
```

(continues on next page)

(продолжение с предыдущей страницы)

```

trace: false
shards:
- master: http://172.20.77.19:8101/v2/
  slave: ''
- master: http://172.20.77.19:8102/v2/
  slave: ''

```

```
sudo docker container restart findface-multi-findface-sf-api-1
```

7. Перезапустите контейнер `findface-multi-findface-multi-legacy-1`, чтобы применились миграции.

```
sudo docker container restart findface-multi-findface-multi-legacy-1
```

Настройка сетевого взаимодействия

После развертывания компонентов FindFace Multi настройте их взаимодействие по сети. Выполните следующие действия:

1. Откройте файл конфигурации `/opt/findface-multi/configs/findface-sf-api/findface-sf-api.yaml`:

```
sudo vi /opt/findface-multi/configs/findface-sf-api/findface-sf-api.yaml
```

Задайте следующие параметры:

Параметр	Описание
<code>extraction-api</code> <code>-> extraction-api</code>	IP-адрес и слушающий порт сервера, являющегося <i>шлюзом</i> для серверов экстракции с настроенной балансировкой нагрузки.
<code>storage-api</code> <code>-> shards -> master</code>	IP-адрес и порт мастера шарда <code>findface-tarantool-server</code> . Остальные шарды прописываются по аналогии.
<code>upload_url</code>	Путь в WebDAV nginx, по которому в компонент <code>findface-upload</code> будут отправляться исходные изображения, миниатюры и нормализованные изображения объектов.

```

...
extraction-api:
  extraction-api: http://172.168.1.9:18667
...
webdav:
  upload-url: http://127.0.0.1:3333/uploads/
...
storage-api:
  ...
  shards:
    - master: http://172.168.1.9:8101/v2/

```

(continues on next page)

(продолжение с предыдущей страницы)

```

slave: ''
- master: http://172.168.1.9:8102/v2/
  slave: ''
- master: http://172.168.1.12:8101/v2/
  slave: ''
- master: http://172.168.1.12:8102/v2/
  slave: ''
- master: http://172.168.1.13:8102/v2/
  slave: ''
- master: http://172.168.1.13:8102/v2/
  slave: ''

```

Перезапустите контейнер `findface-multi-findface-sf-api-1`.

```
sudo docker container restart findface-multi-findface-sf-api-1
```

- Откройте файл конфигурации `/opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py`.

```
sudo vi /opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py
```

Задайте следующие параметры:

Параметр	Описание
<code>SERVICE_EXTERNAL_ADDRESS</code>	FindFace Multi, являющимся приоритетным для Genetec и вебхуков. Если параметр не задан, система использует для работы с данным функционалом <code>EXTERNAL_ADDRESS</code> . Для использования Genetec и вебхуков обязательно укажите по крайней мере один из параметров <code>SERVICE_EXTERNAL_ADDRESS</code> / <code>EXTERNAL_ADDRESS</code> .
<code>EXTERNAL_ADDRESS</code>	(Альтернативно) IP-адрес или URL, который используется для доступа к веб-интерфейсу FindFace Multi. Если параметр не задан, система автоматически определяет его как внешний IP-адрес. Для доступа в FindFace Multi вы можете использовать оба IP-адреса: как автоопределенный, так и указанный в <code>EXTERNAL_ADDRESS</code> .
<code>VIDEO_DETECTOR_TOKEN</code>	Токен и укажите его в данном параметре, чтобы авторизовать модуль видеодетекции объектов.
<code>VIDEO_MANAGER_ADDRESS</code>	Адрес сервера <code>findface-video-manager</code> .
<code>NTLS_HTTP_URL</code>	Адрес сервера <code>findface-ntls</code> .
<code>ROUTER_URL</code>	Внешний IP-адрес сервера <code>findface-security</code> , который будет получать обнаруженные объекты от экземпляра(ов) <code>findface-video-worker</code> .
<code>SF_API_ADDRESS</code>	Адрес сервера <code>findface-sf-api</code> .

```
sudo vi /opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py
```

```

...
# SERVICE_EXTERNAL_ADDRESS is prioritized for FFSecurity webhooks and Genetec
  ↪ plugin.
SERVICE_EXTERNAL_ADDRESS = 'http://localhost'
EXTERNAL_ADDRESS = 'http://127.0.0.1'
...

```

(continues on next page)

(продолжение с предыдущей страницы)

```
FFSECURITY = {
  'VIDEO_DETECTOR_TOKEN': '7ce2679adfc4d74edcf508bea4d67208',
  ...
  'VIDEO_MANAGER_ADDRESS': 'http://127.0.0.1:18810',
  ...
  'NTLS_HTTP_URL': 'http://127.0.0.1:3185',
  'ROUTER_URL': 'http://172.168.1.9',
  ...
  'SF_API_ADDRESS': 'http://127.0.0.1:18411',
  ...
}
```

Перезапустите контейнер `findface-multi-findface-multi-legacy-1`.

```
sudo docker container restart findface-multi-findface-multi-legacy-1
```

На этом установка FindFace Multi в многосерверной среде будет завершена.

Важно: Для того чтобы сохранить совместимость FindFace Multi со средой установки, настоятельно рекомендуется отключить автоматическое обновление Ubuntu. В этом случае вы сможете обновлять ОС вручную, контролируя процесс обновления отдельных пакетов.

Для отключения автоматического обновления Ubuntu выполните следующие команды:

```
sudo apt-get remove unattended-upgrades
sudo systemctl stop apt-daily.timer
sudo systemctl disable apt-daily.timer
sudo systemctl disable apt-daily.service
sudo systemctl daemon-reload
```

Важно: Сервисы FindFace Multi логируют большой объем данных, что в конечном итоге может привести к чрезмерной загрузке диска. Чтобы этого не произошло, рекомендуется отключить сервис `rsyslog` из-за его неоптимальной схемы ротации логов и вместо него использовать правильно настроенный сервис `systemd-journal`. См. пошаговые инструкции в разделе *Логирование*.

2.3.9 Установка моделей нейронных сетей

Для обнаружения и распознавания объектов и их атрибутов `findface-extraction-api` использует нейронные сети.

Если необходим ручной запуск установки моделей, используйте консольный инсталлятор следующим образом:

1. Запустите файл `findface-*.run`.

Примечание: Вместо `findface-*` укажите актуальное имя файла.

```
sudo ./findface-*.run
```

2. Устанавливаемый продукт: FindFace Multi
3. Тип установки: Fully customized installation.
4. Выберите устанавливаемый компонент FindFace Multi: `findface-data`. Для этого сначала снимите выделение со всех компонентов, введя в командной строке `-*`, затем введите порядковый номер компонента. Введите `done` для сохранения выбора и перехода к следующему шагу.
5. Выберите модели для установки. После этого процесс установки будет автоматически запущен.

Вы можете найти установленные модели для распознавания объектов и их атрибутов в директории `/opt/findface-multi/models/`. См. *Обзор моделей нейронных сетей*.

2.3.10 Процедуры и первичные инструкции после развертывания

После завершения развертывания FindFace Multi выполните описанные ниже процедуры.

В этом разделе:

- *Задание временной зоны*
- *Лицензирование экземпляра продукта*
- *Настройка логирования*
- *Полезные команды Docker*

Задание временной зоны

Временная зона определяет время в отчетах, логах, именах артефактов FindFace Multi, таких как полные кадры, миниатюры событий и т. п.

Временная зона указывается в файле конфигурации `/opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py` в формате Регион/Страна/Город или Etc/GMT+Н. Лучший способ сделать это — скопировать и вставить часовой пояс из [таблицы](#) в Википедии.

```
sudo vi /opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py

# time zone
TIME_ZONE = 'America/Argentina/Buenos_Aires'
```

Перезапустите контейнер `findface-multi-findface-multi-legacy-1`.

```
sudo docker container restart findface-multi-findface-multi-legacy-1
```

Лицензирование экземпляра продукта

FindFace Multi предоставляет несколько вариантов лицензирования. Какой бы вариант вы ни выбрали, вам потребуется загрузить файл лицензии FindFace Multi через основной веб-интерфейс `http://<Host_IP_address>/#/license`. Для доступа к веб-интерфейсу используйте учетные данные `superuser`.

Важно: Не передавайте данные `superuser` (Супер Администратора) третьим лицам. Для администрирования системы создайте назначаемого администратора. Отличие назначаемого администратора от Супер Администратора в том, что последний не может лишиться прав администратора даже при смене роли.

Обратитесь к разделу *Лицензирование*, чтобы узнать о доступных вариантах лицензирования.

Настройка логирования

По умолчанию процессы FindFace Multi регистрируются в *логах контейнеров Docker*, доступ к которым можно получить с помощью команд `docker logs` и `docker service logs`. Кроме этого, Docker использует драйвер перевода логов в *json-файлы*, который кэширует логи контейнеров в формате JSON. Вы можете настроить Docker на использование другого драйвера логов, выбрав из *множества доступных механизмов логирования*. См. *Логирование*, чтобы узнать, как это сделать.

Полезные команды Docker

Для быстрого и эффективного администрирования FindFace Multi вы должны обладать обширными знаниями и навыками работы с Docker. Если вы новичок в Docker, начните с приведенных ниже команд. Затем изучите документацию Docker для получения дополнительных навыков.

- Просмотр всех контейнеров Docker, включая остановленные:

```
docker ps -a
```

Чтобы получить более компактный и понятный вывод, используйте следующую команду:

```
docker ps -a --format "table {{.ID}}\t{{.Names}}\t{{.Status}}\t{{.State}}"
```

Чтобы расширить предыдущий вывод, пригодится следующая команда:

```
docker ps --format='{{json .}}' | jq
```

- Перезапуск Docker:

```
sudo systemctl restart docker
```

- Просмотр логов контейнера, если *включен* драйвер перевода логов в `journald`:

```
journalctl CONTAINER_NAME=findface-multi-findface-multi-legacy-1 -f
```

- Остановка контейнера Docker:

```
sudo docker container stop <container_name>/<container_id>
```

Остановка всех контейнеров Docker:

```
sudo docker container stop $(sudo docker ps -a -q)
```

- Запуск контейнера Docker:

```
sudo docker container start <container_name>/<container_id>
```

Запуск всех контейнеров Docker:

```
sudo docker container start $(sudo docker ps -a -q)
```

- Просмотр файла конфигурации docker-compose.yaml для FindFace Multi:

```
cat /opt/findface-multi/docker-compose.yaml
```

- Файлы конфигурации FindFace Multi можно найти здесь:

```
cd /opt/findface-multi/configs/
```

После внесения изменений в файл конфигурации перезапустите соответствующий контейнер, выполнив команду:

```
sudo docker container restart <container_name>/<container_id>
```

- Вход в работающий контейнер Docker для выполнения в нем команды:

```
sudo docker container exec -it <container_name> /bin/bash
```

- Остановка и удаление всех контейнеров FindFace Multi:

```
cd /opt/findface-multi
sudo docker-compose down
```

- Сборка, воссоздание и запуск контейнеров FindFace Multi:

```
cd /opt/findface-multi
sudo docker-compose up -d
```

2.3.11 Удаление экземпляра продукта

FindFace Multi можно удалить, выполнив определенные команды или запустив скрипт.

Важно: Обязательно создайте *резервную копию* вашего экземпляра перед его удалением, если вы планируете позже *восстановить* FindFace Multi и его данные.

Для удаления экземпляра FindFace Multi выполните следующие команды, соблюдая последовательность:

```
cd /opt/findface-multi
sudo docker-compose down
sudo docker system prune -a
cd ~
sudo rm -rf /opt/findface-multi
```

Удалить FindFace Multi можно другим способом, а именно, посредством запуска следующего скрипта:

```
sudo chmod +x /opt/findface-multi/uninstall.sh
cd /opt/findface-multi/
./uninstall.sh /opt/findface-multi/
```

2.4 Администрирование и основная конфигурация

2.4.1 Лицензирование

В этой главе:

- *Принцип лицензирования*
- *Просмотр и обновление лицензии*
- *Лицензирование через USB-ключ аппаратной защиты*
- *Лицензирование по слепку с оборудования*

Принцип лицензирования

FindFace Multi лицензируется по следующим критериям:

1. Общее количество извлеченных векторов признаков, независимо от типа объекта (лицо, силуэт, транспортное средство).

Примечание: Векторы признаков извлекаются из обнаруженных на видео объектов, из фотографий в карточках и фотографий пользователей, при построении т. н. центроидов в процессе формирования кластеров.

Схема лицензирования выглядит следующим образом:

- События: 1 событие распознавания объекта на видео = 1 объект в лицензии.
 - Карточка: 1 фотография в карточке = 1 объект в лицензии.
 - Кластеры: 1 человек = 1 объект в лицензии.
 - Пользователи: 1 фотография пользователя = 1 объект в лицензии.
2. Количество источников видеоизображения, используемых в данный момент времени (т. е. количество активных на данный момент задач на обработку видео с камер и из архивов).
 3. Количество экземпляров моделей, используемых **findface-extraction-api**.
 4. Распознавание атрибутов лица: пол/возраст/эмоции/очки/борода/медицинская маска/др.
 5. Распознавание атрибутов силуэтов: цвет/тип одежды/др.
 6. Распознавание атрибутов автомобилей: марка/модель/цвет/тип кузова/др.
 7. Распознавание номера транспортного средства.
 8. Определение витальности лиц (Liveness).

9. Видеозапись.
10. Интеграции с партнерами.
11. Интеграции с внешними VMS.

Вы можете выбрать между следующими методами лицензирования:

- Онлайн-лицензирование выполняется через Глобальный менеджер лицензий NtechLab `license.ntechlab.com` и требует наличия стабильного интернет-соединения, DNS и открытого порта 443 TCP. После отключения от интернета система продолжит работать в автономном режиме в течение четырех часов.

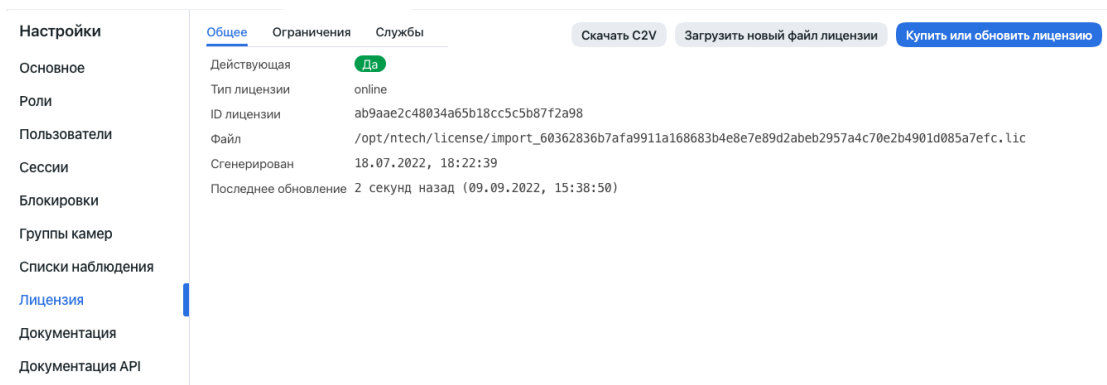
Примечание: Уведомите своего менеджера, если требуется продлить данный период (до 2-х дней).

- Для лицензирования в закрытой сети через USB-ключ необходимо наличие USB-порта на физическом сервере с компонентом `findface-ntls` (сервер лицензирования в составе *ядра FindFace*).
- Для лицензирования в закрытой сети по слепку с оборудования необходимо наличие установленных драйверов Sentinel на физическом сервере с компонентом `findface-ntls`.

Важно: Для обеспечения функционирования системы достаточно одного экземпляра `findface-ntls`. Если ваша система нуждается в большем количестве серверов лицензирования, заблаговременно сообщите об этом своему менеджеру NtechLab, чтобы предотвратить блокировку системы.

Просмотр и обновление лицензии

После установки FindFace Multi загрузите в систему полученный от менеджера файл лицензии. Для этого перейдите в *Настройки* -> *Лицензия*.



Используйте ту же вкладку для просмотра текущей информации по лицензии и обновления лицензии.

Лицензирование через USB-ключ аппаратной защиты

Для того чтобы реализовать лицензирование через USB-ключ, выполните следующие действия:

1. Сообщите своему менеджеру о том, что намерены использовать данный метод лицензирования, и запросите свой USB-ключ и файл лицензии.
2. Откройте файл конфигурации `/opt/findface-multi/docker-compose.yaml`.

```
sudo vi /opt/findface-multi/docker-compose.yaml
```

3. Добавьте строку `privileged: true`. Смонтируйте каталог `/dev` в контейнер `findface-multi-findface-ntls-1`, указав его в томах (volumes) секции `findface-ntls`. В результате секция будет выглядеть следующим образом:

```
findface-ntls:
  command: [--config=/etc/findface-ntls.cfg]
  image: docker.int.ntl/ntech/universe/ntls:ffserver-8.221216
  network_mode: service:pause
  privileged: true
  restart: always
  user: root
  volumes: ['./configs/findface-ntls/findface-ntls.yaml:/etc/findface-ntls.cfg:ro',
    './data/findface-ntls:/ntech/license', '/dev:/dev']
```

4. Создайте новое правило `udev`.
 1. Загрузите файл `95-grdnt.rules` в любой каталог, например, в `/home/username/tmp/`.
 2. Скопируйте файл `95-grdnt.rules` в каталог `/etc/udev/rules.d/`.

```
sudo cp /home/username/tmp/95-grdnt.rules /etc/udev/rules.d/
```

5. Выполните пересборку контейнеров FindFace Multi.

```
cd /opt/findface-multi

sudo docker-compose down

sudo docker-compose up -d
```

6. Вставьте USB-ключ в USB-порт.
7. Загрузите файл лицензии на вкладке *Лицензия*.

Лицензирование по слепку с оборудования

Примечание: Sentinel — вид офлайн лицензий, которые, в отличие от guardant лицензий, не требуют физического носителя для своей работы.

Словарь:

- Sentinel — система защиты ПО и лицензирования от [Thales](#). Она позволяет реализовать офлайн-лицензирование без доступа к глобальному серверу.

- Файл C2V — это файл со слепком с оборудования клиентской машины, который служит для привязки лицензии только в этой машине. Этот файл генерируется библиотекой Sentinel у клиента на той машине, где впоследствии будет устанавливаться лицензионный ключ.

Для того чтобы реализовать лицензирование по слепку, выполните следующие действия:

1. Сообщите своему менеджеру о том, что намерены использовать данный метод лицензирования, и запросите свой уникальный идентификатор лицензии. Менеджер также предоставит вам пакет *findface-sentinel-lib_*.deb*, необходимый для интеграции FindFace Multi с Sentinel.
2. Установите драйверы Sentinel на физическом сервере с компонентом **findface-ntls**.

Выполните следующие действия:

1. Загрузите драйверы Sentinel с официального сайта.
2. Разархивируйте скачанный архив и перейдите в него.

```
tar -xvzf Sentinel_LDK_Linux_Runtime_Installer_script.tar.gz
cd Sentinel_LDK_Linux_Runtime_Installer_script/
```

3. Внутри архива есть еще архив **aksusbd-8.31.1.tar.gz**. Разархивируйте его и перейдите в полученную директорию.

```
tar -xvzf aksusbd-8.31.1.tar.gz
cd aksusbd-8.31.1/
```

4. Выполните команду установки.

```
sudo ./dinst
```

5. Запустите и проверьте статусы сервисов Sentinel.

```
sudo systemctl start aksusbd.service hasplmd.service
sudo systemctl status aksusbd.service hasplmd.service
```

3. Смонтируйте каталоги **/var/hasplm** и **/etc/hasplm** в контейнер **findface-multi-findface-ntls-1**. Для этого откройте файл конфигурации **/opt/findface-multi/docker-compose.yaml** и перечислите их в томах (volumes) секции **findface-ntls**.

```
sudo vi /opt/findface-multi/docker-compose.yaml

findface-ntls:
  ...
  volumes: ['./configs/findface-ntls/findface-ntls.yaml:/etc/findface-ntls.cfg:ro',
↪ './data/findface-ntls:/ntech/license', '/var/hasplm:/var/hasplm', '/etc/hasplm:/
↪ etc/hasplm']
```

4. Выполните пересборку контейнеров FindFace Multi.

```
cd /opt/findface-multi

sudo docker-compose down

sudo docker-compose up -d
```

- Поместите пакет `findface-sentinel-lib_*.deb`, полученный от своего менеджера, в любой каталог на этом же сервере. Установите пакет.

```
sudo dpkg -i /path/to/findface-sentinel-lib_*.deb
```

- В веб-интерфейсе FindFace Multi перейдите на вкладку *Настройки* -> *Лицензия*. Снимите слепок с оборудования (файл C2V), нажав на кнопку *Скачать C2V для активации*.

Совет: Если вы предпочитаете работать с консолью, можно вместо этого отправить следующий API-запрос в `findface-ntls`:

```
curl <findface-ntls-server-ip>:3185/c2v >my_pc.c2v
```

- Отправьте идентификатор лицензии и файл C2V своему менеджеру и получите в ответном письме файл лицензии.
- Загрузите файл лицензии на вкладке *Лицензия*.

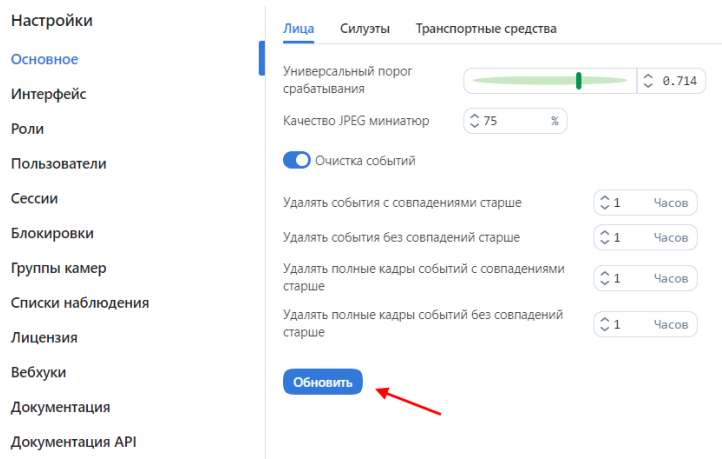
2.4.2 Основные настройки

Основные настройки FindFace Multi определяют работу системы и потребление ресурсов. Доступны следующие настройки:

- универсальный порог срабатывания
- качество JPEG миниатюр
- расписание автоматического удаления старых событий и эпизодов

Настройки для лиц, силуэтов и транспортных средств задаются по отдельности в зависимости от включенных объектов распознавания.

Для изменения основных настроек перейдите *Настройки* -> *Основные*. После внесения изменений нажмите *Обновить*. Подробное описание каждой настройки приведено ниже.



В этом разделе:

- *Универсальный порог срабатывания*
- *Качество JPEG миниатюр*
- *Автоматическое удаление старых событий и эпизодов*

Универсальный порог срабатывания

FindFace Multi принимает решение о совпадении объектов (например, о том, что лица принадлежат одному человеку) на основании предустановленного порога срабатывания. По умолчанию установлено оптимальное пороговое значение. При необходимости вы можете изменить его.

Примечание: Чем выше порог, тем меньше шансов на положительную ложную верификацию человека, однако некоторые подходящие фотографии могут также не пройти верификацию.

Совет: Вы можете настроить порог срабатывания индивидуально для каждой *группы камер* и *списка наблюдения*.

Важно: Для большинства случаев распознавания лиц значение, установленное по умолчанию для универсального порога срабатывания, является оптимальным. Не рекомендуется самостоятельно менять его. Обязательно предварительно проконсультируйтесь с нашими техническими специалистами (support@ntechlab.com).

Качество JPEG миниатюр

В зависимости от качества JPEG, файлы миниатюр могут в конечном счете занять значительную часть дискового пространства. Используйте вкладку *Основные* для настройки параметра.

Автоматическое удаление старых событий и эпизодов

Используйте эту же вкладку, чтобы задать расписание автоматического удаления старых событий и эпизодов из базы данных. Можно настроить удаление событий и эпизодов по разным расписаниям в зависимости от наличия совпадений с картотекой, а также выбрать удаление только полных кадров.

2.4.3 Управление пользователями и безопасность системы

Важно: Хотя FindFace Multi предоставляет инструменты для защиты от несанкционированного доступа, они не заменяют правильно настроенный межсетевой экран. Обязательно используйте фаервол, чтобы усилить защиту сети FindFace Multi.

Управление ролями и пользователями

В этой главе:

- *Предустановленные роли*
- *Создание новой роли*
- *Главная и дополнительная роль пользователя*
- *Создание пользовательского аккаунта вручную*
- *Интеграция с Active Directory для автоматического создания пользователей*
 - *Создание файла Keytab*
 - *Настройка томов для Kerberos и файла keytab*
 - *Настройка NGINX на сервере FindFace Multi для поддержки Active Directory*
 - *Завершение настройки FindFace Multi*
 - *Управление пользователями FindFace Multi через Active Directory*
- *Деактивация или удаление пользователей*

Предустановленные роли

Для работы с FindFace Multi предусмотрены следующие предустановленные роли:

- **Администратор:** предоставляется полный доступ к функциональности FindFace Multi, интеграционным и административным инструментам.

Важно: Супер Администратор не может лишиться прав администратора даже при смене роли.

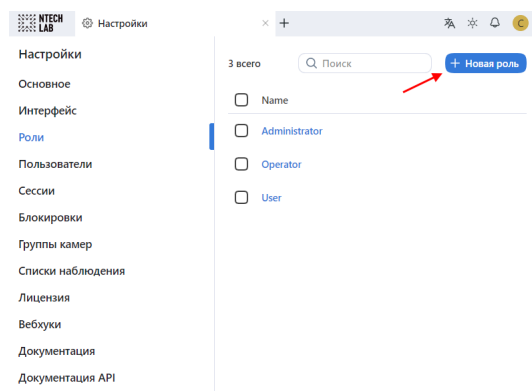
- **Оператор:** предоставляется полный доступ к функциональности FindFace Multi.
- **Пользователь:** обладает правами на работу с событиями и эпизодами, а также на изменение данных в своем профиле. Остальные функции доступны в режиме чтения.

Вы можете изменить привилегии предустановленных ролей, а также создать новые роли.

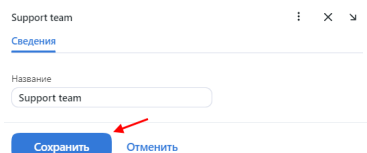
Создание новой роли

Для создания новой роли выполните следующие действия:

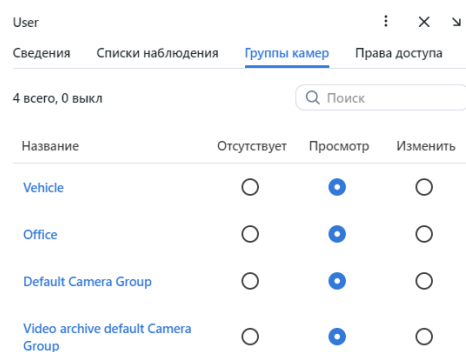
1. Перейдите *Настройки* -> *Роли*.
2. Нажмите *+ Новая роль*.



3. На вкладке *Сведения* задайте имя роли. Сохраните роль.



4. После сохранения роли вы увидите, что рядом с вкладкой *Сведения* появились следующие вклад-ки:



- *Списки наблюдения*: права роли на определенные списки наблюдения
- *Группы камер*: права роли на определенные группы камер
- *Права доступа*: привилегии роли на работу с системными функциями и сущностями

Задайте права роли в соответствии со своими потребностями. Обратите внимание на разницу между правами на определенный список наблюдения/группу камер и правами на системную сущность `watchlist/cameragroup`. Например, если вы установите *Отсутствует* для определенной группы камер на вкладке *Группы камер*, пользователи с соответствующей ролью не смогут

работать с **данной** группой камер. Снятие всех флажков для сущности cameragroup на вкладке *Права доступа* не позволит пользователям просматривать и работать со **всеми** группами камер.

Полный список сущностей FindFace Multi, которые используются в текущей версии:

- **all_own_sessions**: все *сессии* текущего пользователя на разных устройствах

Примечание: Если для данной сущности установлены соответствующие разрешения, пользователи смогут просматривать (*Просмотр*) и закрывать (*Удалить*) все свои сессии на разных устройствах. В противном случае пользователям будет разрешено только просматривать и закрывать свою сессию на текущем устройстве. Работа с сессиями производится на вкладке *Сессии (Настройки)*.

- **bodycluster**: *кластер силуэтов*
- **bodyevent**: *событие распознавания силуэта*
- **bodyobject**: *фотография в карточке человека в полный рост*
- **camera**: *камера*
- **cameragroup**: *группа камер*
- **carcard**: *карточка транспортного средства*
- **carcluster**: *кластер изображений транспортного средства*
- **carepisode**: *эпизод с транспортным средством*
- **carevent**: *событие распознавания транспортного средства*
- **carobject**: *фотография транспортного средства в карточке*
- **counter**: *счетчики*
- **deviceblacklistrecord**: *список блокировок*
- **facecluster**: *кластер лиц*
- **faceevent**: *событие распознавания лица*
- **faceobject**: *фотография лица в карточке*
- **humancard**: *карточка человека*
- **humanepisode**: *эпизод с человеком*
- **report**: *отчеты*
- **upload**: *элемент (фото) в пакетной загрузке*
- **uploadlist**: *список фотографий в пакетной загрузке*
- **user**: *пользователь*
- **videoarchive**: *идентификация объектов в видеофайлах*
- **watchlist**: *список наблюдения*
- **webhook**: *вебхук*

Вы также можете включать и отключать права на следующую функциональность:

- **batchupload_cards**: *пакетная загрузка фотографий*
- **change_runtime_setting**: *изменение основных настроек FindFace Multi*

- `view_auditlog`: просмотр и работа с *аудит логами*.
- `configure_ntls`: конфигурация *сервера лицензий findface-ntls*
- `view_runtime_setting`: просмотр *основных настроек* FindFace Multi

5. Сохраните изменения.

Главная и дополнительная роль пользователя

Вы можете назначить пользователю привилегии, используя следующие роли:

- *Главная роль*: основная роль пользователя, обязательная для назначения. Пользователю можно назначить только одну главную роль.
- *Роль*: дополнительная роль пользователя, необязательная для назначения. Одному пользователю можно назначить несколько ролей. Связанные с ними права будут добавлены к правам, предоставляемым главной ролью.

Все пользователи, принадлежащие к определенной основной роли, автоматически получают доступ к группам камер (и архивам внутри группы) и спискам наблюдения (и карточкам в этих списках), созданным пользователем с той же основной ролью, с учетом привилегий, определенных их дополнительными ролями.

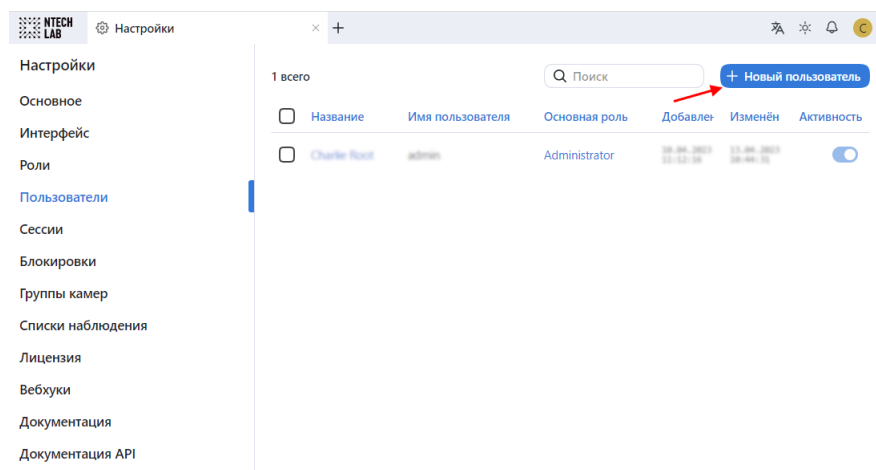
См.также:

Создание пользовательского аккаунта вручную

Создание пользовательского аккаунта вручную

Для создания пользовательского аккаунта вручную выполните следующие действия:

1. Перейдите *Настройки* -> *Пользователи*.
2. Нажмите *+ Новый пользователь*.



3. На вкладке *Сведения* введите данные пользователя, такие как имя, логин и пароль. При необходимости добавьте комментарий.
4. Из раскрывающегося списка *Роли* выберите одну или несколько пользовательских ролей. Назначьте одну из них основной.

5. На вкладке *Фотографии* прикрепите фотографию пользователя.
6. Сохраните аккаунт пользователя.

user_nina
Нина Иванова

Изменено X

Сведения Фотографии

Название
user_nina

Имя пользователя
Нина Иванова

Пароль
.....

Повторите пароль
.....

Комментарий
usermina

Роли	Основная роль
User	<input checked="" type="radio"/>
FBI	<input type="radio"/>

Добавить роль

Сохранить Отменить

Интеграция с Active Directory для автоматического создания пользователей

При большом количестве пользователей в FindFace Multi создание их аккаунтов по одному может быть неудобным. Одним из способов облегчить создание пользователей является использование интеграции FindFace Multi с Active Directory. Для настройки интеграции руководствуйтесь приведенными ниже пошаговыми инструкциями, соблюдая последовательность действий.

Создание файла Keytab

Войдите на сервер Active Directory и выполните следующие действия:

1. Создайте новый аккаунт пользователя в домене Active Directory для использования в качестве сервисного аккаунта.

Выполните следующие действия:

1. Откройте Active Directory. Нажмите *Пуск -> Средства администрирования -> Пользователи и компьютеры Active Directory*.
2. Нажмите на имя домена, а затем разверните содержимое. Щелкните правой кнопкой мыши *Пользователи*, выберите *Создать -> Пользователь*. Отобразится форма создания пользователя.
3. Заполните поля формы на свое усмотрение. На второй вкладке поставьте флажок *Срок действия пароля не ограничен*.
4. Нажмите *Далее*. Проверьте введенные данные и, если все верно, нажмите *Готово*.
5. Щелкните правой кнопкой мыши по созданному аккаунту пользователя, а затем перейдите *Свойства -> Член групп -> Добавить*.
6. Добавьте в список группы *Администраторы домена* и *Пользователи домена*, а затем нажмите «ОК».

7. Нажмите *ОК* для завершения.
2. Зарегистрируйте имя субъекта-службы (Service Principal Name, SPN) для созданного сервисного аккаунта. Для этого откройте PowerShell от имени администратора и выполните следующую команду, указав фактическое имя пользователя (вместо SERVICE USER NAME) и домен. В примере ниже имя домена testntl.local.

```
setspn -A HTTP/<SERVICE USER NAME>.testntl.local@TESTNTL.LOCAL <SERVICE USER NAME>
```

3. В том же окне PowerShell сгенерируйте keytab-файл, выполнив приведенную ниже команду с фактическим именем сервисного пользователя SERVICE USER NAME, доменом и желаемым именем файла вместо KEYTAB FILE NAME.

```
ktpass.exe -princ HTTP/<SERVICE USER NAME>.testntl.local@TESTNTL.LOCAL -mapuser  
↪<SERVICE USER NAME> -crypto ALL -ptype KRB5_NT_PRINCIPAL -pass * -out c:\<KEYTAB_  
↪FILE NAME>.keytab
```

Чтобы проверить результат, перейдите в корневую директорию диска C. В ней вы увидите keytab-файл с соответствующим именем.

4. Переместите созданный файл keytab на сервер FindFace Multi.

Настройка томов для Kerberos и файла keytab

Для успешного установления связи между FindFace Multi и Active Directory необходимо включить поддержку Kerberos в контейнере findface-multi-findface-multi-ui-1 на основном сервере FindFace Multi. Выполните следующие действия:

1. Смонтируйте файл /opt/findface-multi/configs/kerberos/krb5.conf и директорию /opt/findface-multi/configs/keytab в контейнер findface-multi-findface-multi-ui-1. Для этого откройте файл конфигурации /opt/findface-multi/docker-compose.yaml и перечислите их в томах (volumes) секции findface-multi-ui.

```
sudo vi /opt/findface-multi/docker-compose.yaml

findface-multi-ui:
  ...
  volumes: [ './configs/findface-multi-ui/nginx-site.conf:/etc/nginx/conf.d/default.  
↪conf:ro',  
            './data/findface-multi-legacy/uploads:/var/lib/findface-security/uploads',  
            './configs/kerberos/krb5.conf:/etc/krb5.conf:ro',  
            './data/findface-multi-ui/keytab/:/keytab/' ]
```

2. Создайте директории для смонтированных томов: /opt/findface-multi/configs/kerberos/ и /opt/findface-multi/data/findface-multi-ui/keytab/. Скопируйте файл конфигурации Kerberos из контейнера findface-multi-findface-multi-ui-1 в директорию /opt/findface-multi/configs/kerberos/.

```
sudo mkdir -p /opt/findface-multi/data/findface-multi-ui/keytab/  
sudo mkdir /opt/findface-multi/configs/kerberos/  
sudo docker cp findface-multi-findface-multi-ui-1:/etc/krb5.conf /opt/findface-  
↪multi/configs/kerberos/
```

3. Откройте файл конфигурации /opt/findface-multi/configs/kerberos/krb5.conf. Укажите realm Active Directory в секции libdefaults. Имя realm должно совпадать с доменным име-

нем Active Directory, но быть написано в верхнем регистре (TESTNTL.LOCAL в приведенном ниже примере). Также укажите домен Active Directory в секции `realms` по аналогии с примером ниже.

```
sudo vi /opt/findface-multi/configs/kerberos/krb5.conf

[libdefaults]
    default_realm = TESTNTL.LOCAL
...
[realms]
TESTNTL.LOCAL = {
    kdc = testntl.local
    default_domain = testntl.local
}
```

4. Скопируйте файл `keytab` в директорию `/opt/findface-multi/data/findface-multi-ui/keytab/`.
5. Добавьте следующую строку в файл `/etc/hosts`: `<IP-адрес сервера Active Directory> <имя домена Active Directory>`.

```
vi /etc/hosts

...
192.168.0.5 testntl.local
```

Настройка NGINX на сервере FindFace Multi для поддержки Active Directory

1. Откройте файл конфигурации `/opt/findface-multi/configs/findface-multi-ui/nginx-site.conf`. Найдите раздел `location /users/me/ad` и раскомментируйте его. Заполните секцию по аналогии с примером ниже, указав свои актуальные переменные в строки с комментариями (`#`).

Переменные, которые необходимо указать, следующие:

- `auth_gss_realm`: имя `realm` в Kerberos
- `auth_gss_keytab`: путь к `keytab`-файлу.
- `auth_gss_service_name`: полное имя сервисного пользователя, включая имя домена, к которому он принадлежит

```
sudo vi /opt/findface-multi/configs/findface-multi-ui/nginx-site.conf

location /users/me/ad {
    proxy_pass http://127.0.0.1/auth/ad_login/; # e.g http://127.0.0.1/auth/ad_
↪login/;
    proxy_method POST;

    proxy_set_header    X-Real-IP $remote_addr;
    proxy_set_header    Host $http_host;
    proxy_set_header    X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header    Authorization $http_authorization;
    proxy_pass_header    Authorization;
    proxy_no_cache 1;
    proxy_cache_bypass 1;

    auth_gss on;
```

(continues on next page)

(продолжение с предыдущей страницы)

```

auth_gss_realm TESTNTL.LOCAL; # e.g. TESTNTL.LOCAL;
auth_gss_keytab /keytab/user.keytab; # e.g. /var/lib/web.keytab
auth_gss_service_name HTTP/user.testntl.local; # e.g. HTTP/web.testntl.local;
auth_gss_allow_basic_fallback on;
}

```

Завершение настройки FindFace Multi

Чтобы завершить интеграцию FindFace Multi с Active Directory, выполните на стороне FindFace Multi следующие действия по настройке:

1. Откройте файл конфигурации `/opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py`.

```
sudo vi /opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py
```

2. В разделе SERVICES установите `"active_directory": True`.

```

SERVICES = {
    ...
    "active_directory": True,
    ...
}
}

```

3. Заполните раздел `ACTIVE_DIRECTORY_CONFIG` следующим образом:

- `AUTH_LDAP_SERVER_URI`: ldap: <IP-адрес сервера Active Directory>
- `AUTH_LDAP_BIND_DN`: имя сервисного пользователя, созданного в Active Directory
- `AUTH_LDAP_BIND_PASSWORD`: пароль сервисного пользователя
- `SEARCH_GROUPS`: организационные единицы Active Directory, в которых FindFace Multi будет выполнять поиск аккаунтов пользователей

```

# Specify server credentials
ACTIVE_DIRECTORY_CONFIG = {
    'AUTH_LDAP_SERVER_URI': 'ldap://192.168.0.5',
    # Domain Administrator user
    'AUTH_LDAP_BIND_DN': '<SERVICE USER NAME IN ACTIVE DIRECTORY>',
    # Domain Administrator user password
    'AUTH_LDAP_BIND_PASSWORD': 'SERVICE USER NAME PASSWORD',
    # Specify organization units where users search will be executed.
    # Follow pattern (e.g. OU=DEV,DC=domain,DC=com)
    'SEARCH_GROUPS': 'OU=DEV,DC=testntl,DC=local',
}

```

4. Откройте файл конфигурации `/opt/findface-multi/configs/findface-multi-identity-provider/findface-multi-identity-provider.py` и повторите предыдущие шаги.
5. Выполните пересборку контейнеров FindFace Multi.

```
cd /opt/findface-multi

sudo docker-compose down

sudo docker-compose up -d
```

Управление пользователями FindFace Multi через Active Directory

Если интеграция FindFace Multi с Active Directory включена, вы сможете задать одну из групп Active Directory для создаваемой или редактируемой роли.

3 всего

☐ Name

☒ Administrator

☐ Operator

☐ User

Administrator

Сведения | Списки наблюдения | Группы камер | Права доступа →

Название

Administrator

Группа Active Directory

Admins

ID 1

Как только пользователь из выбранной группы Active Directory впервые войдет в FindFace Multi, он будет автоматически добавлен в список пользователей FindFace Multi.

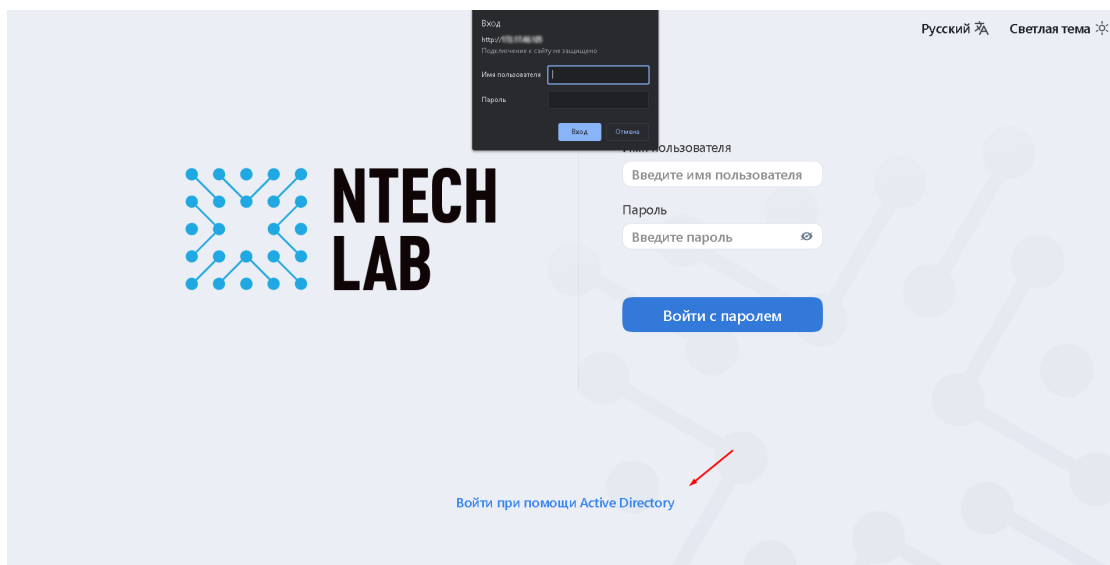
2 всего

Поиск

+ Новый пользователь

<input type="checkbox"/> Название	Имя пользователя	Основная роль	Active Directory	Добавлен	Изменён	Активность
<input type="checkbox"/> test8	test8	Administrator	Да	19.04.2023 18:28:10	19.04.2023 18:28:10	<input checked="" type="checkbox"/>
<input type="checkbox"/> Charlie Root	admin	Administrator		19.04.2023 16:24:43	19.04.2023 18:29:55	<input type="checkbox"/>

Чтобы войти в FindFace с помощью Active Directory, пользователь должен нажать кнопку *Войти при помощи Active Directory* в окне аутентификации, указать свои учетные данные Active Directory и нажать *Sign in*.



Деактивация или удаление пользователей

Для того чтобы деактивировать пользователя, выключите переключатель *Активность* в списке пользователей (*Настройки* -> *Пользователи*).

Если вы собираетесь деактивировать нескольких пользователей, выберите их в списке и нажмите *Деактивировать выбранные*.

Для того чтобы удалить пользователей из FindFace, выберите их в списке и нажмите *Удалить выбранные*.

Аутентификация и мониторинг сессии пользователя

В этом разделе:

- *Типы аутентификации*
- *Настройка аутентификации и мониторинга сессии*
- *Принудительный вывод всех пользователей из системы*

Типы аутентификации

В FindFace Multi возможны следующие типы аутентификации:

- **password:** стандартная аутентификация посредством логина и пароля. Включена по умолчанию.
- **face:** аутентификация возможна только по лицу пользователя.
- **face_or_password:** аутентификация возможна по лицу или с логином/паролем.
- **face_and_password:** двухфакторная аутентификация. После успешного распознавания лица пользователь должен ввести свои логин и пароль.

Важно: Для всех типов аутентификации, основанных на распознавании лиц, необходима следующая конфигурация:

- автономный сервис liveness (`findface-liveness-api`)
 - *HTTPS*
-

Важно: Перед использованием распознавания лиц для аутентификации *прикрепите фотографии* к профилям пользователей и оборудуйте их рабочие места вебкамерами.

Примечание: Вы можете включить мониторинг рабочей сессии для типов аутентификации `face` и `face_or_password`. В этом случае система будет периодически возобновлять сессию после проверки, что лицо человека на рабочем месте совпадает с лицом вошедшего в систему пользователя (подробнее см. в *Настройка аутентификации и мониторинга сессии*).

Совет: FindFace Multi также предоставляет независимо настраиваемую аутентификацию на основании сертификата. Обратитесь к нашим экспертам за консультацией (support@ntechlab.com).

Настройка аутентификации и мониторинга сессии

Для настройки аутентификации и мониторинга сессии пользователя выполните следующие действия:

1. Откройте файл конфигурации `/opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py`. Найдите секции `FFSECURITY` и `FFSECURITY_AUTH_CONFIG`.

```
sudo vi /opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py

FFSECURITY = {
    # auth config
    # available options: face, password, face_and_password, face_or_password
    'AUTH_TYPE': 'face_or_password',
    # 180 days by default
    'MAXIMUM_SESSION_LENGTH': 15552000,
    ...
}

...
# - FindFace Security authorization configuration dictionary -

FFSECURITY_AUTH_CONFIG = {
    'FACE_AUTH_CONFIDENCE': 0.740, # FAR = 2.5E-09 # model: [kiwi_320]
    # 3 settings below are for front-end only
    # session renew works only with face or face_or_password authorization type
    'NEED_SESSION_RENEW': False,
    'RENEW_SESSION_INTERVAL': 0,
    'MAXIMUM_RENEW_ATTEMPTS': 2,
}
```

2. В разделе FFSECURITY задайте следующие параметры аутентификации:

- AUTH_TYPE: тип аутентификации. Доступные варианты: `face`, `password`, `face_and_password`, `face_or_password`.
- MAXIMUM_SESSION_LENGTH: максимальная продолжительность сессии пользователя в секундах. По истечении времени сессии происходит принудительный вывод пользователя из системы при условии, что сессия не была предварительно возобновлена.

3. В разделе FFSECURITY_AUTH_CONFIG задайте следующие параметры аутентификации и мониторинга сессии:

- FACE_AUTH_CONFIDENCE: после того как лицо на видео с вебкамеры распознано как живое, система сравнивает его с фотографиями пользователей, используя для верификации данную пороговую степень схожести.
- NEED_SESSION_RENEW: если `True`, сессия пользователя может быть возобновлена и продлена на время, равное `MAXIMUM_SESSION_LENGTH`, после проверки, что лицо человека на рабочем месте совпадает с лицом вошедшего в систему пользователя.
- RENEW_SESSION_INTERVAL: период в секундах до ожидаемого времени завершения сессии, в течение которого система попытается возобновить сессию, активировав вебкамеру для верификации лица пользователя.
- MAXIMUM_RENEW_ATTEMPTS: количество попыток верификации пользователя. Попытки выполняются одна за другой в течение интервала возобновления сессии.

Примечание: Попытка верификации занимает приблизительно 3 секунды.

Совет: Мы рекомендуем настроить параметры мониторинга таким образом, что величина `MAXIMUM_RENEW_ATTEMPTS`, умноженная на продолжительность попытки, была меньше значения `RENEW_SESSION_INTERVAL`. В противном случае система увеличит интервал возобновления сессии в 2, 3 и более раз, в зависимости от количества попыток.

4. Перезапустите контейнер `findface-multi-findface-multi-legacy-1`.

```
sudo docker container restart findface-multi-findface-multi-legacy-1
```

Принудительный вывод всех пользователей из системы

Для того чтобы принудительно вывести всех пользователей из системы, выполните следующую команду в консоли центрального сервера FindFace Multi:

```
sudo docker exec findface-multi-findface-multi-legacy-1 /opt/findface-security/bin/
↪python3 /tigre_prototype/manage.py logout_all_users
```

Совет: Данная команда пригодится при массовом переходе к другому типу аутентификации.

Настройка шифрования данных

Для обеспечения безопасности данных включите SSL-шифрование. Выполните следующие действия:

1. На хосте создайте директорию с конфигурацией nginx, включающую каталог для хранения информации о SSL-шифровании:

```
sudo mkdir -p /etc/nginx/ssl/
```

2. Создайте ключ и сертификат SSL. Если будет использоваться самоподписанный сертификат, выполните следующую команду:

```
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/nginx/ssl/my-  
example-domain.com.key -out /etc/nginx/ssl/my-example-domain.com.crt
```

Для заполнения полей сертификата вам будет предложено несколько вопросов. Ответьте на них, уделив особое внимание строке **Common Name**. В ней нужно ввести имя или публичный IP-адрес домена, связанного с сервером. Созданные файлы ключа `my-example-domain.com.key` и сертификата `my-example-domain.com.crt` будут сохранены в каталоге `/etc/nginx/ssl/`.

3. Если используется подтвержденный сертификат, то нужно добавить путь сертификата в **volumes** для сервиса `findface-video-worker`, добавить установку `ca-certificates` и обновить корневое хранилище сертификатов в контейнере сервиса.

1. Откройте файл `docker-compose.yaml`:

```
sudo vi /opt/findface-multi/docker-compose.yaml
```

2. Найдите раздел `findface-video-worker` и отредактируйте его, чтобы он выглядел так, как показано ниже.

Для CPU:

```
findface-video-worker:
  entrypoint: ["sh", "-c", "apt-get update && DEBIAN_FRONTEND=noninteractive &&  
apt-get install --no-install-recommends --yes ca-certificates && update-ca-  
certificates && exec /tini -- /findface-video-worker-cpu --config=/etc/  
findface-video-worker.yaml"]
  depends_on: [findface-video-manager, findface-ntls, mongoddb]
  image: docker.int.ntl/ntech/universe/video-worker-cpu:ffserver-8.221216
  logging: {driver: journald}
  network_mode: service:pause
  restart: always
  volumes: ['./configs/findface-video-worker/findface-video-worker.yaml:/etc/  
findface-video-worker.yaml:ro',  
            './models:/usr/share/findface-data/models:ro', './cache/findface-video-  
worker/models:/var/cache/findface/models_cache',  
            './cache/findface-video-worker/recorder:/var/cache/findface/video-worker-  
recorder',  
            '/etc/nginx/ssl/my-example-domain.crt:/usr/local/share/ca-certificates/my-  
example-domain.crt:ro']
```

Для GPU достаточно будет добавить путь до сертификата и обновить корневое хранилище:

```
findface-video-worker:
  entrypoint: ["sh", "-c", "update-ca-certificates && exec /tini -- /findface-
```

(continues on next page)

(продолжение с предыдущей страницы)

```

↪video-worker-gpu --config=/etc/findface-video-worker.yaml"]
  depends_on: [findface-video-manager, findface-ntls, mongodb]
  environment: [CUDA_VISIBLE_DEVICES=0]
  image: docker.int.ntl/ntech/universe/video-worker-gpu:ffserver-8.221216
  logging: {driver: journald}
  network_mode: service:pause
  restart: always
  runtime: nvidia
  volumes: ['./configs/findface-video-worker/findface-video-worker.yaml:/etc/
↪findface-video-worker.yaml:ro',
    './models:/usr/share/findface-data/models:ro', './cache/findface-video-
↪worker/models:/var/cache/findface/models_cache',
    './cache/findface-video-worker/recorder:/var/cache/findface/video-worker-
↪recorder',
    '/etc/nginx/ssl/my-example-domain.crt:/usr/local/share/ca-certificates/my-
↪example-domain.crt:ro']

```

Важно: Для CPU версии настройка требует доступ к интернету. Если он отсутствует, обратитесь к нашим техническим специалистам (support@ntechlab.com).

Предупреждение: Для CPU версии вырастет время запуска контейнера `findface-video-worker` на ~15 секунд.

3. Выполните пересборку контейнеров FindFace Multi.

```

cd /opt/findface-multi/
docker-compose down
docker-compose up -d

```

4. Настройте nginx для использования SSL. Откройте файл конфигурации nginx `/opt/findface-multi/configs/findface-multi-ui/nginx-site.conf`. Внесите в файл следующие изменения:

1. Добавьте новый раздел `server {...}`, содержащий правило замены URL. В строке `rewrite ^(.*) https://...` замените `ip_address_server_ffmulti` IP-адресом сервера, на котором установлен экземпляр FindFace Multi.

```

server {
    listen 80;
    server_name my-example-domain.com www.my-example-domain.com;
    rewrite ^(.*) https://ip_address_server_ffmulti$1 permanent;
    access_log off;
}

```

2. Закомментируйте следующие строки в существующем разделе `server {...}`:

```

# listen 80 default_server;
# listen [::]:80 default_server;

```

3. Добавьте следующие строки, включая пути к сертификату и ключу, в существующий раздел `server {...}`:

```
listen 443 ssl;

ssl_certificate      /etc/nginx/ssl/my-example-domain.com.crt;
ssl_certificate_key  /etc/nginx/ssl/my-example-domain.com.key;
```

Пример файла конфигурации `/opt/findface-multi/configs/findface-multi-ui/nginx-site.conf` с корректно заданными настройками SSL показан в примере ниже:

```
upstream ffsecurity {
    server 127.0.0.1:8002;
}

upstream ffsecurity-ws {
    server 127.0.0.1:8003;
}

upstream ffsecurity-django {
    server 127.0.0.1:8004;
}

upstream audit {
    server 127.0.0.1:8012;
}

upstream identity-provider {
    server 127.0.0.1:8022;
}

map $http_upgrade $ffsec_upstream {
    default "http://ffsecurity-ws";
    "" "http://ffsecurity";
}

server {
    listen 80;
    server_name my-example-domain.com www.my-example-domain.com;
    rewrite ^(.*) https://my-example-domain.com$1 permanent;
    access_log off;
}

server {
    # listen 80 default_server;
    # listen [::]:80 default_server;

    listen 443 ssl;
    ssl_certificate /etc/nginx/ssl/my-example-domain.com.crt;
    ssl_certificate_key /etc/nginx/ssl/my-example-domain.com.key;

    root /var/lib/findface-security;

    autoindex off;
```

(continues on next page)

(продолжение с предыдущей страницы)

```

server_name _;

location = / {
    alias /usr/share/findface-security-ui/;
    try_files /index.html =404;
}
location /static/ {
}
location /uploads/ {
    # internal; # uncomment if you intend to enable OVERPROTECT_
MEDIA
    add_header 'Access-Control-Allow-Origin' '*';
    add_header 'Access-Control-Allow-Methods' '*';
    add_header 'Access-Control-Allow-Headers' '*';
    add_header 'Access-Control-Expose-Headers' 'Content-Length,
Content-Range';
    add_header 'Access-Control-Max-Age' 2592000;

    location ~ /card/(?<card_type>[a-zA-Z]+)/(?<card_id>[0-9]+)/
attachments/(.*)$ {
        add_header 'Access-Control-Allow-Origin' '*';
        add_header 'Access-Control-Allow-Methods' '*';
        add_header 'Access-Control-Allow-Headers' '*';
        add_header 'Access-Control-Expose-Headers' 'Content-
Length,Content-Range';
        add_header 'Access-Control-Max-Age' 2592000;
        add_header 'Content-Disposition' 'attachment';
        add_header 'Content-Security-Policy' 'sandbox';
    }
}
location /ui-static/ {
    alias /usr/share/findface-security-ui/ui-static/;
}
location /doc/ {
    alias /opt/findface-security/doc/;
}
location /api-docs {
    alias /opt/findface-security/rapidoc;
    index index.html;
}
location /api-docs/ {
    alias /opt/findface-security/rapidoc/;
    try_files $uri index.html =404;
}
location ~ /videos/(?<video_id>[0-9]+)/upload/(.*)$ {
    client_max_body_size 15g;

    proxy_set_header Host $http_host;
    proxy_set_header X-Forwarded-For $remote_addr;
    proxy_set_header X-Forwarded-Proto $scheme;
    proxy_pass http://ffsecurity;
}

```

(continues on next page)

(продолжение с предыдущей страницы)

```

        location @django {
            internal;
            client_max_body_size 1g;
            proxy_set_header Host $http_host;
            proxy_set_header X-Forwarded-For $remote_addr;
            proxy_set_header X-Forwarded-Proto $scheme;
            proxy_http_version 1.1;
            proxy_read_timeout 5m;
            proxy_pass http://ffsecurity-django;
        }

#       location /v1/video-liveness {
#           add_header Access-Control-Allow-Headers "*" always;
#           add_header Access-Control-Allow-Methods "*" always;
#           add_header Access-Control-Allow-Origin "*" always;
#
#           if ($request_method = 'OPTIONS') {
#               return 204;
#           }
#
#           client_max_body_size 300m;
#           proxy_set_header Host $http_host;
#           proxy_set_header X-Forwarded-For $remote_addr;
#           proxy_set_header X-Forwarded-Proto $scheme;
#           proxy_pass http://127.0.0.1:18301;
#           proxy_read_timeout 5m;
#       }

        location / {
            client_max_body_size 1g;
            proxy_set_header Host $http_host;
            proxy_set_header X-Forwarded-For $remote_addr;
            proxy_set_header X-Forwarded-Proto $scheme;
            proxy_http_version 1.1;
            proxy_set_header Upgrade $http_upgrade;
            proxy_set_header Connection "upgrade";
            proxy_pass $ffsec_upstream;
            proxy_read_timeout 5m;

            location ~ ^/(cameras|videos|vms|external-vms).*/stream/?$ {
                proxy_set_header Host $http_host;
                proxy_set_header X-Forwarded-For $remote_addr;
                proxy_set_header X-Forwarded-Proto $scheme;
                proxy_pass http://ffsecurity;
            }

            location ~ ^/streams/(.*)$ {
                internal;
                proxy_pass $1$is_args$args;
            }

            location /audit-logs {

```

(continues on next page)

(продолжение с предыдущей страницы)

```

        proxy_pass http://audit;
    }

    location ~ ^/(auth/ad_
groups|cproauth|groups|permissions|sessions|users|user-face|device-blacklist-
records) {
        proxy_pass http://identity-provider;
    }
}
# location /users/me/ad {
#
#         proxy_pass <FFmulti_address>/auth/ad_login/; e.g http://127.0.
0.1/auth/ad_login/;
#         proxy_method POST;
#
#         proxy_set_header    X-Real-IP $remote_addr;
#         proxy_set_header    Host $http_host;
#         proxy_set_header    X-Forwarded-For $proxy_add_x_forwarded_for;
#         proxy_set_header    Authorization $http_authorization;
#         proxy_pass_header Authorization;
#         proxy_no_cache 1;
#         proxy_cache_bypass 1;
#
#         auth_gss on;
#         auth_gss_realm <REALM>; # e.g. TESTNTL.LOCAL;
#         auth_gss_keytab <path/to/file.keytab>; # e.g. /var/lib/web.
keytab
#         auth_gss_service_name <service_name>; # e.g. HTTP/web.testntl.
local;
#         auth_gss_allow_basic_fallback on;
#     }
}

```

4. Скопируйте общий файл конфигурации nginx `nginx.conf` из контейнера `findface-multi-findface-multi-ui-1` в директорию `/etc/nginx/`:

```

sudo docker cp findface-multi-findface-multi-ui-1:/etc/nginx/nginx.conf /etc/
nginx/nginx.conf

```

5. В общем файле конфигурации `/etc/nginx/nginx.conf` найдите раздел `SSL Settings` и добавьте в его конец следующие строки:

```

ssl_session_cache    shared:SSL:10m;
ssl_session_timeout 1h;

```

5. В файле `/opt/findface-multi/docker-compose.yaml` добавьте (смонтируйте) каталог, хранящий данные SSL-шифрования `/etc/nginx/ssl/` и файл конфигурации `/etc/nginx/nginx.conf` системы-хоста в контейнер `findface-multi-findface-multi-ui-1`:

1. Откройте файл `docker-compose.yaml`:

```

sudo vi /opt/findface-multi/docker-compose.yaml

```

- Найдите раздел `findface-multi-ui` и отредактируйте его, чтобы он выглядел так, как показано в примере ниже:

```
findface-multi-ui:
  depends_on: [findface-multi-legacy]
  image: docker.int.ntl/ntech/multi/multi/ui:ffmulti-2.0.0
  network_mode: service:pause
  restart: always
  volumes: ['./configs/findface-multi-ui/nginx-site.conf:/etc/nginx/conf.d/
  ↪default.conf:ro',
    './data/findface-multi-legacy/uploads:/var/lib/findface-security/uploads',
    '/etc/nginx/ssl:/etc/nginx/ssl',
    '/etc/nginx/nginx.conf:/etc/nginx/nginx.conf:ro']
```

- Отредактируйте файл конфигурации `opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py`.

- В параметрах `ROUTER_URL` и `IMAGE_CROP_URL` замените префикс `http://` на `https://`.

```
sudo vi /opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.
  ↪py

...
'ROUTER_URL': 'https://127.0.0.1',
'IMAGE_CROP_URL': 'https://127.0.0.1',
...
```

- При использовании подтвержденного сертификата необходимо в `ROUTER_URL` указать домен, по которому был создан сертификат:

```
'ROUTER_URL': 'https://my-example-domain.com'
```

- Добавьте адрес `https://my-example-domain.com` в параметр `EXTERNAL_ADDRESS`:

```
...
EXTERNAL_ADDRESS = 'https://my-example-domain.com'
...
```

- На сервере с установленным FindFace Multi откройте файл `/etc/hosts` и добавьте следующую строку:

```
sudo vi /etc/hosts

...
127.0.0.1 my-example-domain.com
```

- В системе, где вы используете браузер для взаимодействия с FindFace Multi, откройте файл `hosts`. Добавьте IP-адрес сервера, на котором установлен FindFace Multi, вместо `ip_address_server_ffmulti`. Замените `my-example-domain.com` адресом вашего домена — так же, как вы делали в предыдущих шагах.

- Для операционной системы Linux выполните следующее:

```
sudo vi /etc/hosts
```

(continues on next page)

(продолжение с предыдущей страницы)

```
...
*ip_address_server_ffmulti* my-example-domain.com
```

2. Если вы используете операционную систему Windows, откройте файл C:\Windows\System32\drivers\etc\hosts от имени администратора. Добавьте следующую строку в файл hosts:

```
*ip_address_server_ffmulti* my-example-domain.com
```

9. Перезапустите контейнеры:

```
cd /opt/findface-multi/
sudo docker-compose down
sudo docker-compose up -d
```

10. При использовании самоподписанного сертификата для работы камер и загруженных видеоархивов нужно отключать проверку SSL-сертификата:

1. Перейдите на вкладку *Источники видео* -> *Камеры* или *Загрузки*.
2. Нажмите на камеру или загруженный видеоархив.
3. На вкладке *Дополнительные* снимите флажок *Проверить SSL-сертификат*.

Предупреждение: Если среда будет сильно распределена при *полностью настраиваемой установке* и компонент `pause` будет исключен из установки, то могут возникнуть проблемы с получением скриншотов в камерах и в работе с видеоплеером. Для решения этой проблемы можно добавить следующие строки в файл `/opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py`:

```
USE_X_FORWARDED_HOST = True
SECURE_PROXY_SSL_HEADER = ('HTTP_X_FORWARDED_PROTO', 'https')
```

Перезапустите контейнер `findface-multi-findface-multi-legacy-1`.

```
sudo docker restart findface-multi-findface-multi-legacy-1
```

Включение защиты картотеки

Если защита картотеки отключена, фотографии и вложения карточек будут доступны по прямой ссылке независимо от прав пользователя. Для того чтобы повысить безопасность картотеки, настройте FindFace Multi на выполнение всех медиа-запросов через приложение DJANGO для дополнительной проверки ACL.

Важно: Используйте защиту содержимого карточек только при необходимости, поскольку данная настройка оказывает серьезное негативное влияние на производительность системы.

Важно: Чтобы проверка ACL выполнялась правильно, необходимо установить разрешение на просмотр загруженных в карточки фотографий лиц, силуэтов и транспортных средств. Для этого перейдите *Настройки* → *Роли* → нужная роль → *Права доступа* и установите *Просмотр* для сущностей `faceobject`, `bodyobject` и `carobject`, в зависимости от того, фотографии каких объектов есть в карточках. Подробнее см. *Создание новой роли*.

См. также:

[Картотека](#)

Для включения защиты картотеки выполните следующие действия:

1. Откройте файл конфигурации `/opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py`. Найдите параметр `OVERPROTECT_MEDIA` и установите его в значение `True`.

```
sudo vi /opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py

...

'OVERPROTECT_MEDIA': True,
```

2. Сделайте то же самое в файле конфигурации `/opt/findface-multi/configs/findface-multi-identity-provider/findface-multi-identity-provider.py` : найдите параметр `OVERPROTECT_MEDIA` и установите его в значение `True`.

```
sudo vi /opt/findface-multi/configs/findface-multi-identity-provider/findface-multi-identity-provider.py

...

'OVERPROTECT_MEDIA': True,
```

3. Откройте файл конфигурации `nginx` `/opt/findface-multi/configs/findface-multi-ui/nginx-site.conf`. Раскомментируйте параметр `internal` в разделе `location /uploads`.


```
sudo vi /opt/findface-multi/configs/findface-multi-ui/nginx-site.conf

location /uploads/ {
    internal; # Uncomment if you intend to enable OVERPROTECT_MEDIA
    ...
}
```

4. Перезапустите контейнеры `findface-multi-findface-multi-legacy-1`, `findface-multi-findface-multi-identity-provider-1` и `findface-multi-findface-multi-ui-1`.

```
sudo docker container restart findface-multi-findface-multi-legacy-1
sudo docker container restart findface-multi-findface-multi-identity-provider-1
sudo docker container restart findface-multi-findface-multi-ui-1
```

5. После применения новой политики безопасности вошедшие в систему пользователи должны пройти повторную аутентификацию. Для того чтобы они сделали это, принудительно выведите их из системы командой:

```
sudo docker container exec -it findface-multi-findface-multi-identity-provider-1 /
↪opt/findface-security/bin/python3 /tigre_prototype/manage.py logout_all_users
```

Отключение ACL

Постоянные проверки разрешений потребляют большой объем системных ресурсов. При необходимости ACL FindFace Multi можно отключить.

Выполните следующие действия:

1. Откройте файл конфигурации `/opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py`.

```
sudo vi /opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py
```

2. Установите `ENABLE_ACL = False`.

```
...

ENABLE_ACL = False
```

3. Перезапустите контейнер `findface-multi-findface-multi-legacy-1`.

```
sudo docker container restart findface-multi-findface-multi-legacy-1
```

Список сессий пользователей. Блокировки

В этой главе:

- Предоставление разрешений для работы с сессиями
- Просмотр сессий пользователей
- Блокировка устройства

FindFace Multi позволяет увидеть текущие пользовательские сессии и узнать связанные с ними данные, такие как UUID подключенного устройства, тип пользовательского интерфейса, IP-адрес, время последнего соединения и т. д.

При необходимости вы можете добавить устройство в список заблокированных, при этом учетная запись пользователя останется активной. Блокировка устройства может пригодиться в различных ситуациях, например, если вы хотите, чтобы пользователи работали с системой только со своих рабочих мест. Данные меры помогут вам поднять безопасность системы на новый уровень.

Предоставление разрешений для работы с сессиями

Доступ пользователей к списку сессий зависит от предоставленных *разрешений*:

- Администратор: может просматривать и закрывать сессии всех пользователей
- Пользователь с разрешениями `all_own_sessions`: может просматривать/закрывать все сессии со своего логина
- Пользователь без разрешений `all_own_sessions`: может просмотреть/закрыть только свою текущую сессию

Просмотр сессий пользователей

Для просмотра списка пользователей перейдите *Настройки -> Сессии*.

UUID	Пользователь	Пользовательский интерфейс	IP-адрес	Статус	Последнее соединение
ee99d19-8e88-44b7-a834-176eaf63efaa	admin	{ "user_agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/15.6 Safari/605.1.15" }	172.28.78.46	Онлайн	09.09.2022 15:50:54
006b4028-c761-472f-9f26-1ff3e24ad424	admin		172.28.78.46	Офлайн	09.09.2022 12:42:59
702fb88f-c8ec-4469-90fa-bdb8b25bd4e9	admin		172.28.78.10	Офлайн	09.09.2022 12:32:29

Каждая запись сессии содержит следующие данные:

- UUID устройства
- логин
- тип пользовательского интерфейса (мобильный/веб)
- информация об устройстве
- IP-адрес
- статус (онлайн, офлайн, заблокировано)
- время последнего соединения

Используйте панель фильтров сверху, чтобы задать условия поиска в списке сессий.

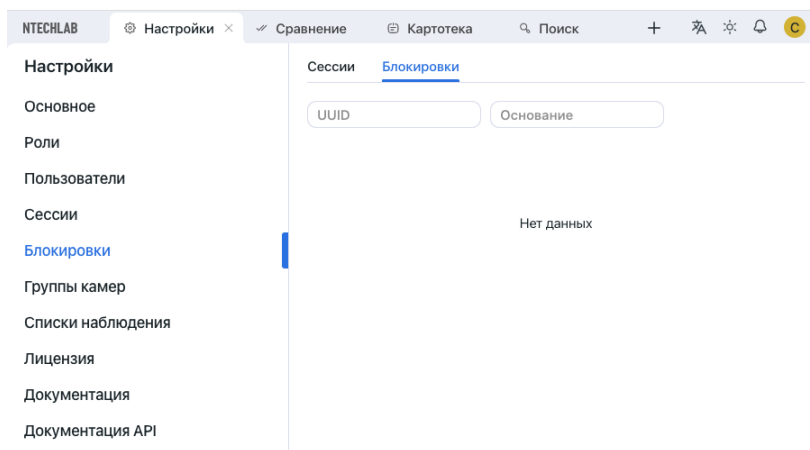
Для закрытия сессии выберите ее в списке и нажмите *x*.

UUID	Пользователь	Устройство	IP	Статус	Последнее соединение
<input type="checkbox"/> b27c9087-c4f8-4af0-9ba6-3e069b81362f	admin		172.20.78.74	online	30.08.2022 12:23:28
<input checked="" type="checkbox"/> e34c4c6f-51f1-4322-ad0b-203e86f3f59d	admin		172.20.78.30	offline	29.08.2022 17:04:35
<input checked="" type="checkbox"/> 5928916b-ffe7-4507-b6dd-2be826d761aa	admin		172.20.78.114	offline	24.08.2022 15:18:14
<input checked="" type="checkbox"/> acd85d21-9417-4ced-943c-0d6efd727993	admin		172.20.78.102	offline	12.08.2022 18:10:15
<input checked="" type="checkbox"/> f422f69a-913f-469f-9b7c-5db452a296a9	admin		172.20.78.50	offline	08.08.2022 15:56:14
<input checked="" type="checkbox"/> 93c8c7bf-6aa7-4a8b-98be-afb21e542309	admin		172.20.78.6	offline	05.08.2022 18:26:04
<input checked="" type="checkbox"/> 09b30e09-6e00-4d41-8ec5-cc90be5ea69d	admin		172.20.78.62	offline	03.08.2022 20:27:04
<input checked="" type="checkbox"/> 710b4708-6ac9-4c9e-abe3-96566329efbc	admin		172.20.78.106	offline	03.08.2022 19:35:00
<input type="checkbox"/> 3f5abdd-bd4e-4cbe-9ff5-a9b7868c7707	admin		172.20.78.62	offline	03.08.2022

☒ 7 ☐
Блокировать

Блокировка устройства

Список заблокированных устройств доступен на вкладке *Блокировки*.



Вы можете заблокировать устройство на вкладке *Сессии*. Блокировка устройства приводит к автоматическому выводу пользователя из системы.

Для блокировки устройства выполните следующие действия:

1. Выберите соответствующие записи сессий.
2. Нажмите *Блокировать*.

UUID	Пользователь	Устройство	IP	Статус	Последнее соединение
<input type="checkbox"/> b27c9087-c4f8-4af0-9ba6-3e069b81362f	admin		192.168.1.100	online	30.08.2022 12:23:28
<input checked="" type="checkbox"/> e34c4c6f-51f1-4322-ad0b-203e86f3f59d	admin		192.168.1.100	offline	29.08.2022 17:04:35
<input checked="" type="checkbox"/> 5928916b-ffe7-4507-b6dd-2be826d761aa	admin		192.168.1.100	offline	24.08.2022 15:18:14
<input checked="" type="checkbox"/> acd85d21-9417-4ced-943c-0d6efd727993	admin		192.168.1.100	offline	12.08.2022 18:10:15
<input checked="" type="checkbox"/> f422f69a-913f-469f-9b7c-5db452a296a9	admin		192.168.1.100	offline	08.08.2022 15:56:14
<input checked="" type="checkbox"/> 93c8c7bf-6aa7-4a8b-98be-afb21e542309	admin		192.168.1.100	offline	05.08.2022 18:26:04
<input checked="" type="checkbox"/> 09b30e09-6e00-4d41-8ec5-cc90be5ea69d	admin		192.168.1.100	offline	03.08.2022 20:27:04
<input checked="" type="checkbox"/> 710b4708-6ac9-4c9e-abe3-96566329efbc	admin		192.168.1.100	offline	03.08.2022 19:35:00
<input type="checkbox"/> 3f5abbdd-bd4e-4cbe-9ff5-a9b7868c7707	admin		192.168.1.100	offline	03.08.2022

3. Укажите основание для блокировки устройства (обязательно) и дату окончания срока действия блокировки (опционально). Если дата не указана, блокировка будет постоянной.

4. Нажмите *Сохранить*.

Создание блокировки

UUID

e34c4c6f-51f1-4322-ad0b-203e86f3f59d, 5928916b-ffe7-4507-b6dd-2be826d761aa, acd85d21-9417-4ced-943c-0d6efd727993, f422f69a-913f-469f-9b7c-5db452a296a9, 93c8c7bf-6aa7-4a8b-98be-afb21e542309, 09b30e09-6e00-4d41-8ec5-cc90be5ea69d, 710b4708-6ac9-4c9e-abe3-96566329efbc

Основание

Основание

Истекает

15.06.2031 12:00

Отмена

Сохранить

Разрешенные расширения файлов в карточках

По умолчанию вы можете прикрепить к *карточке* файл с любым расширением. Существует возможность повысить безопасность системы, создав список разрешенных расширений файлов. Это предотвратит загрузку пользователями файлов нежелательных форматов, в том числе тех, которые могут включать скрытый вредоносный код, например, `.js`, `.swf` и других.

Для того чтобы создать список разрешенных расширений файлов, выполните следующие действия:

- Откройте файл конфигурации `/opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py`.

```
sudo vi /opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py
```

2. В разделе FFSECURITY найдите параметр `CARD_ATTACHMENTS_FILENAME_REGEX`. Задайте выражение, описывающее разрешенные файловые расширения. Подойдет любое действительное [регулярное выражение Python](#).

Примеры:

- `r'.*\.png'`: разрешает только файлы с расширением `.png`
- `r'.*\.(png|jpg)'`: разрешает только расширения `.png` и `.jpg`
- `r'.*'`: разрешает все расширения
- `None`: разрешает все расширения
- `'XXXXXX'`: загрузка файлов с любым расширением запрещена

```
FFSECURITY = {
    ...
    'CARD_ATTACHMENTS_FILENAME_REGEX': r'.*\.txt',
    ...
}
```

Совет: Если закомментировать параметр `CARD_ATTACHMENTS_FILENAME_REGEX`, это также разрешит все расширения.

3. Перезапустите контейнер `findface-multi-findface-multi-legacy-1`.

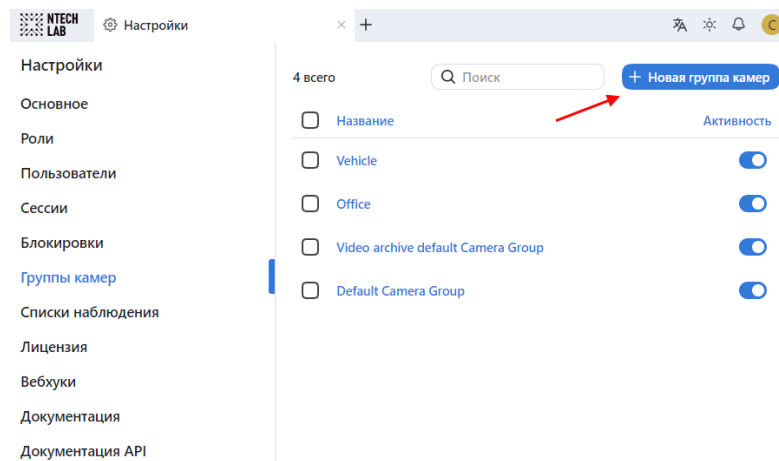
```
sudo docker container restart findface-multi-findface-multi-legacy-1
```

2.4.4 Группы камер

Группы камер — это системные объекты, которые используются для классификации источников видео. После обработки видео система отнесет полученные события распознавания объектов к назначенной группе камер. Это значительно упрощает дальнейшую обработку событий и поиск.

Для создания группы камер выполните следующие действия:

1. Перейдите *Настройки* -> *Группы камер*.
2. Нажмите *+ Новая группа камер*.



3. На вкладке *Информация*, задайте имя группы. При необходимости добавьте комментарий.

4. Если вам нужно выделить определенный экземпляр `findface-video-worker` для обработки видеопотоков с данной группы камер, создайте или выберите из уже созданных одну или несколько меток.

Примечание: Чтобы завершить выделение, перечислите метки в файле конфигурации `/opt/findface-multi/configs/findface-video-worker/findface-video-worker.yaml`. Подробнее см. [Привязка группы камер к экземпляру findface-video-worker](#).

5. Если события от камер, принадлежащих одной группе, требуется дедуплицировать, т. е. исключить одинаковые события, поставьте флажок *Дедуплицировать события с интервалом* и задайте в секундах интервал дедупликации (интервал, с которым события проверяются на уникальность).

Предупреждение: Используйте дедупликацию очень осторожно. Если камеры из одной группы наблюдают разные сцены, некоторые объекты могут быть пропущены. Подробнее см. [Дедупликация событий](#).

6. По умолчанию на всех группах камер в системе используется *универсальный порог срабатывания*. Для того чтобы установить индивидуальный порог для группы камер, включите *Порог срабатывания* и укажите нужное пороговое значение.
7. Сохраните изменения.
8. На вкладке *Права доступа* назначьте права на работу с группой камер, указав, пользователям с какими ролями разрешено изменять/просматривать ее настройки.

Название	Просмотр	Изменить
Administrator	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Operator	<input checked="" type="checkbox"/>	<input type="checkbox"/>
User	<input checked="" type="checkbox"/>	<input type="checkbox"/>

9. Нажмите *Сохранить*.
10. Активируйте группу камер.

2.4.5 Привязка группы камер к экземпляру findface-video-worker

Часто в распределенной архитектуре обработку видеоизображения с группы камер требуется выполнять локально, не обращаясь к центральному серверу и не перераспределяя видеопотоки между удаленными экземплярами `findface-video-worker`.

Примечание: Например, это может быть актуальным для сетей гостиниц, магазинов, при наличии нескольких проходных в одном здании и т. д.

В этом случае группу камер привязывают к локально установленному экземпляру `findface-video-worker`.

Выполните следующие действия:

1. Перейдите на вкладку *Настройки*. Выберите *Группы камер*.
2. Откройте настройки группы камер.
3. В поле *Ярлыки* создайте или выберите из уже созданных одну или несколько меток для привязки группы камер к экземпляру `findface-video-worker`. Сохраните изменения.
4. Откройте файл конфигурации `/opt/findface-multi/configs/findface-video-worker/findface-video-worker.yaml` и перечислите указанные ранее метки в формате словаря (метки `MyLabel1`, `MyLabel2` в примере ниже).

```
sudo vi /opt/findface-multi/configs/findface-video-worker/findface-video-worker.yaml

labels: {"MyLabel1": "true", "MyLabel2": "true"}
```

5. Перезапустите контейнер `findface-multi-findface-video-worker-1`.

```
sudo docker container restart findface-multi-findface-video-worker-1
```

Примечание: Если камере присвоена метка, то видеопоток с нее может обрабатываться как экземпляром `findface-video-worker` с аналогичной меткой, так и экземпляром `findface-video-worker` без меток.

Предупреждение: Если камера с меткой обрабатывается экземпляром `findface-video-worker` без меток и появляется свободный экземпляр с меткой, камера автоматически на него не переключится. Перезапустите камеру с меткой.

2.4.6 Списки наблюдения

Появление определенных людей и транспортных средств на видео отслеживается с помощью набора стандартных и пользовательских списков наблюдения.

Карточки лиц и транспортных средств добавляются в списки наблюдения. После активации списка наблюдения система будет искать каждого человека или транспортное средство из него при обработке видео.

Вы можете создать столько пользовательских списков наблюдения, сколько необходимо: розыск, подозреваемые и т. д. — в зависимости от потребностей.

В этом разделе:

- *Мониторинг незарегистрированных лиц*
- *Создание списка наблюдения*
- *Удаление списка наблюдения*

Мониторинг незарегистрированных лиц

Базовая конфигурация FindFace Multi уже содержит предустановленный список наблюдения, предназначенный для мониторинга событий, для которых отсутствуют совпадения с картотекой. Данный список наблюдения не может быть удален из системы. Для редактирования настроек списка, перейдите на вкладку *Настройки*. Выберите *Списки наблюдения* и щелкните *Unmatched*.

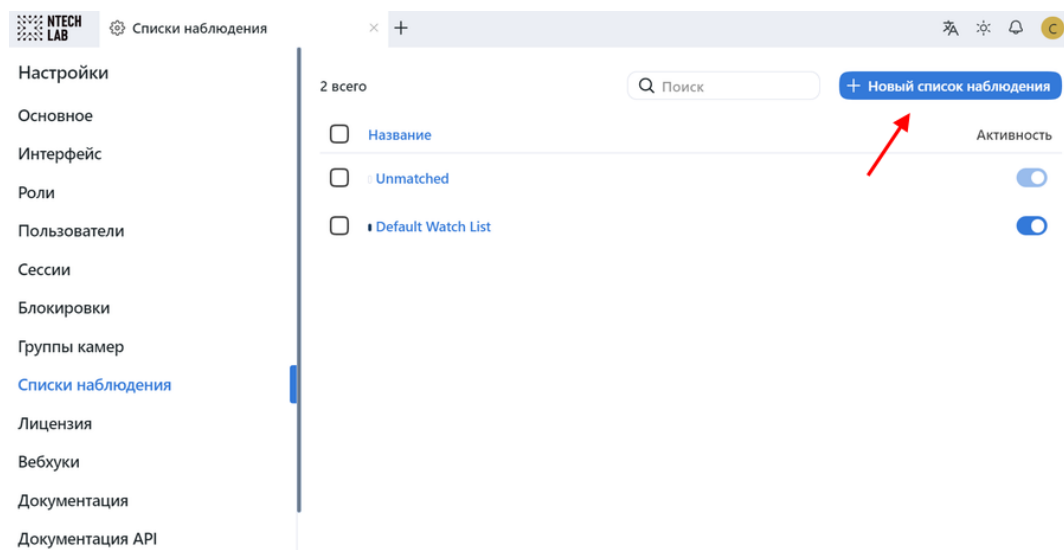
The screenshot displays the 'Unmatched' observation list configuration window. The sidebar on the right has a blue header with a white 'U' and the text 'Unmatched'. The main content area has a title bar with 'Unmatched' and icons for settings, close, and expand. Below the title bar are two tabs: 'Сведения' (selected) and 'Права доступа'. The 'Сведения' tab contains the following settings:

- Название** (Name): A text field containing 'Unmatched'.
- Цвет** (Color): A color picker showing '#ffffff'.
- Группы камер** (Camera groups): A dropdown menu showing 'Не выбраны' (None selected).
- Комментарий** (Comment): A text area containing 'Default list for unmatched events'.
- Порог срабатывания** (Trigger threshold): A toggle switch that is currently turned off.
- Требовать подтверждения событий** (Require event confirmation): An unchecked checkbox.
- Не создавать события** (Do not create events): An unchecked checkbox.
- Включить звуковые оповещения** (Enable sound notifications): An unchecked checkbox.
- Активен** (Active): A checked checkbox.

Создание списка наблюдения

Для создания пользовательского списка наблюдения выполните следующие действия:

1. Перейдите *Настройки* -> *Списки наблюдения*.
2. Нажмите + *Новый список наблюдения*.




3. На вкладке *Сведения* задайте имя списка наблюдения.

4. Из палитры *Цвет* выберите цвет уведомлений для данного списка.

Розыск Изменено X ⋮ X ↗

Сведения

Название Цвет

Розыск  # ff070a

Группы камер

Не выбраны ▼

Комментарий

☐ Порог срабатывания

☒ Требовать подтверждения событий

☐ Не создавать события

☒ Включить звуковые оповещения

☒ Активен

Сохранить Отменить

5. При необходимости опишите назначение списка наблюдения в комментарии.
6. По умолчанию ко всем спискам наблюдения в системе применяется *универсальный порог срабатывания*. Для того чтобы задать индивидуальный порог для списка наблюдения, поставьте флажок *Порог срабатывания* и укажите нужное значение.

Важно: Для большинства случаев распознавания значение порога срабатывания, установленное по умолчанию, является оптимальным. Не рекомендуется самостоятельно менять его. Обязательно предварительно проконсультируйтесь с нашими техническими специалистами (support@ntechlab.com).

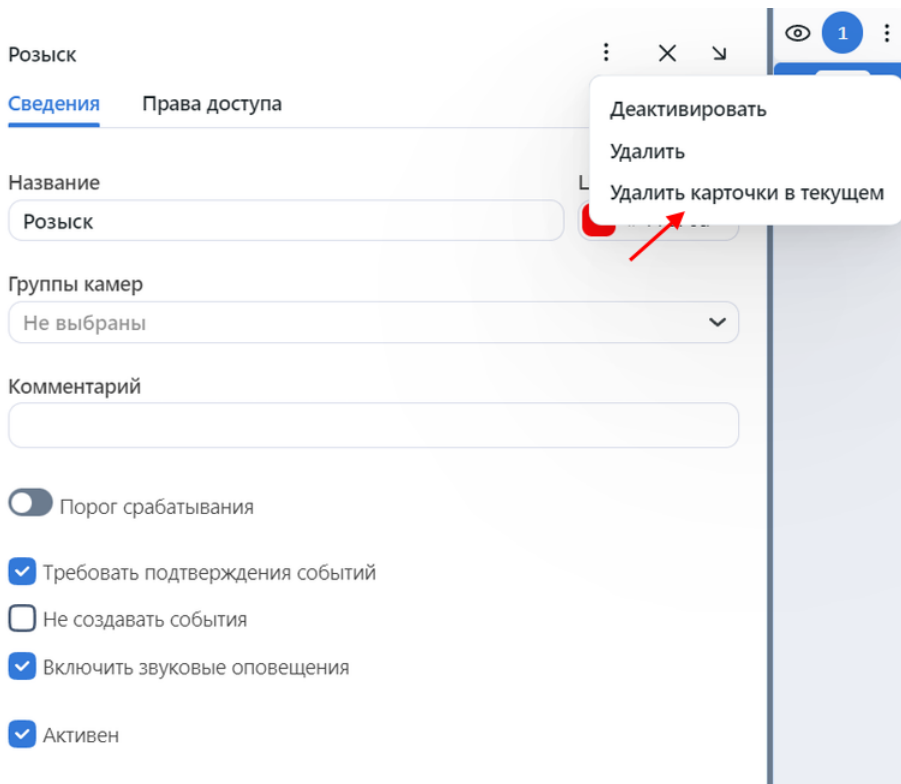
7. Поставьте флажок *Требовать подтверждения событий*, если для данного списка оператор должен в обязательном порядке подтвердить принятие события.
8. Поставьте флажок *Не создавать события*, чтобы не создавать событие, если был обнаружен объект из данного списка.
9. Поставьте флажок *Включить звуковые оповещения*, чтобы включить звук при появлении события для данного списка.
10. На вкладке *Права доступа* назначьте права на список наблюдения, указав роли пользователей, которые смогут изменять/просматривать его настройки.

Розыск	⋮ × ↘	
Сведения	<u>Права доступа</u>	
3 всего	🔍 Поиск	
Название	Просмотр	Изменить
Administrator	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Operator	<input type="checkbox"/>	<input type="checkbox"/>
User	<input type="checkbox"/>	<input type="checkbox"/>

11. Активируйте и сохраните список наблюдения.

Удаление списка наблюдения

Чтобы удалить пользовательский список наблюдения, сначала удалите связанные с ним карточки. В противном случае система выдаст ошибку и не удалит список наблюдения.



Списки наблюдения *Default Watch List* и *Unmatched* не могут быть удалены из системы.

2.4.7 Настройка сохранения изображений в отчетах

При создании *отчетов* предоставляется возможность выбрать способ представления в нем изображений: в виде ссылок, миниатюр или полных кадров. Вы можете настроить параметры изображений. Для этого откройте файл конфигурации `/opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py` и измените заданные по умолчанию качество JPEG и максимальную высоту миниатюр и полных кадров в зависимости от свободного дискового пространства.

```
sudo vi /opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py

# reports image saving options
'REPORT_THUMBNAIL_JPEG_QUALITY': 75,
'REPORT_THUMBNAIL_MAX_HEIGHT': 100,
'REPORT_FULLFRAME_JPEG_QUALITY': 75,
'REPORT_FULLFRAME_MAX_HEIGHT': 250,
```

Перезапустите контейнер `findface-multi-findface-multi-legacy-1`.

```
sudo docker container restart findface-multi-findface-multi-legacy-1
```

2.4.8 Дедупликация событий

В этом разделе:

- *Включение дедупликации*
- *Алгоритм работы дедупликации*

Рассмотрите возможность включения дедупликации, чтобы исключить дублирование событий распознавания объектов в пределах одной группы камер.

Включение дедупликации

Для того чтобы активировать функцию дедупликации, выполните следующие действия:

1. Для каждой камеры в группе включите буферный режим детектирования объектов. См. подробнее *Добавление камеры*.
2. Перейдите на вкладку *Настройки*. Выберите *Группы камер*.
3. Откройте настройки группы камер.
4. Поставьте флажок *Убрать дубли событий с интервалом* и задайте в секундах интервал дедупликации.

Алгоритм работы дедупликации

Алгоритм дедупликации работает следующим образом. В буферном режиме сервер получает один лучший кадр с объектом за всю сессию отслеживания.

Примечание: Сессия отслеживания продолжается до момента исчезновения объекта из поля зрения камеры.

Если в пределах группы камер произошло несколько сеансов отслеживания на одной или нескольких камерах в течение указанного интервала дедупликации, FindFace Multi обработает полученные изображения объектов следующим образом:

- Если в течение предшествующего периода, равного интервалу дедупликации, есть совпадение с карточкой, FindFace Multi отбрасывает вновь полученное изображение. В противном случае изображение сохраняется в базе данных.
- Для объектов без совпадений при выполнении дедупликации FindFace Multi учитывает как сходство между объектами, так и качество кадров. В результате FindFace Multi удаляет все изображения похожих объектов в пределах интервала дедупликации, если они более низкого качества, чем первое в данном интервале. Если новое изображение объекта более высокого качества, оно сохраняется. Это гарантирует, что система дедуплицирует события, не пропуская высококачественные изображения, необходимые для последующей видеоаналитики.

2.4.9 Включение защиты персональных данных

В FindFace Multi реализована поддержка законов, связанных с обработкой персональных данных физических лиц (GDPR и аналогичные).

Для того чтобы применить к системе защиту персональных данных, выполните следующие действия:

1. Откройте файл конфигурации `/opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py`.

```
sudo vi /opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py
```

2. Отключите сохранение событий без совпадений, установив `'IGNORE_UNMATCHED': True`.

```
...

FFSECURITY = {
    ...

    # do not save unmatched events (GDPR support)
    'IGNORE_UNMATCHED': True,

    ...
}
```

3. Для событий с совпадениями включите размытие в полных кадрах всех объектов, для которых совпадения не были найдены. Для этого установите `'BLUR_UNMATCHED_OBJECTS': True`. При желании вы можете изменить установленное по умолчанию для данных кадров качество JPEG.

```
...

FFSECURITY = {
    ...
    # blur all unmatched objects on the full frame of the matched event (GDPR
    ↳support)
    'BLUR_UNMATCHED_OBJECTS': True,

    # full frame jpeg quality when `BLUR_UNMATCHED_OBJECTS` is enabled
    'BLURRED_FULLFRAME_JPEG_QUALITY': 85,

    ...
}
```

4. Включите размытие на *видеостене* всех объектов, для которых совпадения не были найдены. Для этого установите `"gdpr": True` в секции `FFSECURITY_UI_CONFIG -> video_player`.

```
FFSECURITY_UI_CONFIG = {
    ...

    "video_player": {
        "overlay": {
            ...
            "gdpr": True
        }
    }
```

(continues on next page)

(продолжение с предыдущей страницы)

```
}
...
}
```

5. Перезапустите контейнер `findface-multi-findface-multi-legacy-1`.

```
sudo docker container restart findface-multi-findface-multi-legacy-1
```

См.также:

Видеостена

2.4.10 Настройка Видеомэгнитофона

Внутренняя *архитектура* Ядра FindFace позволяет встроить дополнительную функциональность «Видеомэгнитофон», которая позволяет записывать, хранить и проигрывать видеоданные с камер.

Данный раздел посвящен настройке Видеомэгнитофона.

В этом разделе:

- *Включение Видеомэгнитофона*
- *Особенности отключения Видеомэгнитофона*

Включение Видеомэгнитофона

Для активации Видеомэгнитофона выполните следующие действия:

1. Включите возможность передачи видеофрагментов от `findface-video-worker` в сервис `findface-video-storage`.

Важно: Данная настройка делает функциональность Видеомэгнитофона доступной через *HTTP API*. Она также обязательна, если вы собираетесь использовать Видеомэгнитофон в составе *веб-интерфейса* FindFace Multi.

Выполните следующие действия:

1. Откройте файл конфигурации `/opt/findface-multi/configs/findface-video-worker/findface-video-worker.yaml` для включения сервиса `findface-video-worker`, который будет поставлять видео в Видеомэгнитофон.

```
sudo vi /opt/findface-multi/configs/findface-video-worker/findface-video-worker.
↪yaml
```

2. Установите `enabled: true` в разделе `recorder`.

```
recorder:
  enabled: true
...
```

3. Перезапустите контейнер `findface-multi-findface-video-worker-1`.

```
sudo docker container restart findface-multi-findface-video-worker-1
```

2. Настройте Видеомегнитофон для работы в составе веб-интерфейса FindFace Multi.

Примечание: Пропустите следующие шаги, если вам не нужно, чтобы инструменты Видеомегнитофона отображались в веб-интерфейсе FindFace Multi.

Выполните следующие действия:

1. Откройте файл конфигурации `/opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py`.

```
sudo vi /opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py
```

2. Найдите раздел `FFSECURITY_UI_CONFIG -> vms`.

Список параметров для включения элементов видеомегнитофона в интерфейсе FindFace Multi:

Параметр	Описание
<code>vms -> "enabled"</code>	Установите <code>True</code> , чтобы добавить флажок <i>Включить запись</i> в <i>настройки камеры</i> .
<code>vms -> "video_player"</code>	Установите <code>True</code> , чтобы включить <i>отображение видеоплеера</i> в уведомлениях о событиях и <i>предварительном просмотре камеры</i> (вместо статических кадров). Данная настройка требует установленного <code>"enabled": True</code> .

3. Найдите раздел `FFSECURITY_UI_CONFIG -> video_player`.

Список параметров для настройки параметров видеоплеера:

Параметр	Описание
"overlay" -> "objects"	Определяет настройки для отображения рамок и данных по атрибутам для лиц, силуэтов и транспортных средств соответственно. Установите True или False чтобы включить/выключить их отображение.
"overlay" -> "gdpr"	Установите True , чтобы включить размытие на видеоплеере всех объектов без совпадений. Для того чтобы в полном объеме выполнить требования законов о защите персональных данных, руководствуйтесь <i>данной инструкцией</i> .
"timeline" -> "min_zoom"	Определяет самый крупный возможный масштаб на <i>временной шкале видеоплеера</i> , секунды/пиксель.
"timeline" -> "max_zoom"	Определяет самый мелкий возможный масштаб на временной шкале, секунды/пиксель.
"timeline" -> "objects" -> "events" "faces" -> "enabled"	Установите True , чтобы отмечать на временной шкале все события с лицами в виде маркеров.
"timeline" -> "objects" -> "events" "faces" -> "limit"	Максимальное количество событий с лицами, одновременно отмеченных на временной шкале. Если событий с лицами больше, вам будет предложено увеличить масштаб.
"timeline" -> "objects" -> "events" "faces" -> "matchedColor"	Цвет событий с лицами с совпадением.
"timeline" -> "objects" -> "events" "faces" -> "unmatchedColor"	Цвет событий с лицами без совпадений.
"timeline" -> "objects" -> "events" "bodies" -> "enabled"	Установите True , чтобы отмечать на временной шкале все события с силуэтами в виде маркеров.
"timeline" -> "objects" -> "events" "bodies" -> "limit"	Максимальное количество событий с силуэтами, одновременно отмеченных на временной шкале. Если событий с силуэтами больше, вам будет предложено увеличить масштаб.
"timeline" -> "objects" -> "events" "bodies" -> "matchedColor"	Цвет событий с силуэтами с совпадением.
"timeline" -> "objects" -> "events" "bodies" -> "unmatchedColor"	Цвет событий с силуэтами без совпадений.
"timeline" -> "objects" -> "events" "cars" -> "enabled"	Установите True , чтобы отмечать на временной шкале все события с транспортными средствами в виде маркеров.
"timeline" -> "objects" -> "events" "cars" -> "limit"	Максимальное количество событий с транспортными средствами, одновременно отмеченных на временной шкале. Если событий с транспортными средствами больше, вам будет предложено увеличить масштаб.
"timeline" -> "objects" -> "events" "cars" -> "matchedColor"	Цвет событий с транспортными средствами с совпадением.
"timeline" -> "objects" -> "events" "cars" -> "unmatchedColor"	Цвет событий с транспортными средствами без совпадений.


```

FFSECURITY_UI_CONFIG = {

  "video_player": {
    "overlay": {
      "objects": {
        "faces": {
          "bbox": True,
          "info": True
        },
        "bodies": {
          "bbox": True,
          "info": True
        },
        "cars": {
          "bbox": True,
          "info": True
        }
      },
      "gdpr": False
    },
    "timeline": {
      "min_zoom": 0.2,
      "max_zoom": 200,
      "objects": {
        "events": {
          "faces": {
            "enabled": True,
            "limit": 500,
            "matchedColor": "rgba(6,193,103,0.8)",
            "unmatchedColor": "rgba(232,92,74,0.8)"
          },
          "bodies": {
            "enabled": False,
            "limit": 500,
            "matchedColor": "rgba(6,193,103,0.8)",
            "unmatchedColor": "rgba(232,92,74,0.8)"
          },
          "cars": {
            "enabled": False,
            "limit": 500,
            "matchedColor": "rgba(6,193,103,0.8)",
            "unmatchedColor": "rgba(232,92,74,0.8)"
          }
        },
        "episodes": {
          "humans": {
            "enabled": False,
            "limit": 500,
            "matchedColor": "rgba(6,193,103,0.8)",
            "unmatchedColor": "rgba(232,92,74,0.8)"
          },
          "cars": {

```

(continues on next page)

(продолжение с предыдущей страницы)

```

        "enabled": False,
        "limit": 500,
        "matchedColor": "rgba(6,193,103,0.8)",
        "unmatchedColor": "rgba(232,92,74,0.8)",
    },
},
},
},
"vms": {
    "enabled": True,
    "video_player": True,
},

```

4. Перезапустите контейнер `findface-multi-findface-multi-legacy-1`.

```
sudo docker container restart findface-multi-findface-multi-legacy-1
```

Особенности отключения Видеомагнитофона

Если Видеомагнитофон работает и записывает видео на выбранных камерах, а вам нужно отключить его, обязательно сначала отключите запись видео на данных камерах. Только после этого можно перейти к изменению файлов конфигурации `/opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py` и `/opt/findface-multi/configs/findface-video-worker/findface-video-worker.yaml`.

2.4.11 Удаление видеозаписей

См.также:

Настройка Видеомагнитофона

В системе доступно два метода очистки видеоархива:

1. Регулярная очистка
2. Ручная очистка посредством консольной команды

В этом разделе:

- *Регулярная очистка видео*
- *Ручная очистка видео*

Регулярная очистка видео

Для настройки регулярной очистки видео выполните следующие действия:

1. Откройте файл конфигурации `/opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py`.

```
sudo vi /opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py
```

2. Найдите секцию `SERVICES` и установите `"vms_cleanup": True`.

```
SERVICES = {
    "ffsecurity": {
        ...
        "vms_cleanup": True,
    }
}
```

3. Найдите секцию `VMS_CLEANUP_SETTINGS` и укажите, какие именно фрагменты видео должны быть удалены, используя следующие параметры:

- `'CLEANUP_BETWEEN_TRACKS'`: установите `True` для удаления промежутков видео, не содержащих событий распознавания
- `'CLEANUP_ARCHIVE'`: установите `True` для удаления всего видеоархива старше заданного количества дней
- `'ARCHIVE_CLEANUP_AGE'`: максимальный возраст видеоархива в системе, дни. Параметр актуален, если `'CLEANUP_ARCHIVE': True`
- `'CLEANUP_EVENTS_TYPES'`: типы событий, которые будут сохранены в видеоархиве, в форме словаря, например, `[face, body, car]`
- `'KEEP_EVENT_BEFORE_AFTER'`: промежуток до и после трека с событиями, который не будет удален в целях сохранения связанной с ними значимой информации, секунды. Значение должно быть меньше `CLEANUP_THRESHOLD`
- `'CLEANUP_THRESHOLD'`: минимальный промежуток между треками с событиями, секунды. Если интервал между треками короче, он не будет удален

Совет: См. калькулятор `RRULE`.

```
'VMS_CLEANUP_SETTINGS': {
    'CLEANUP_BETWEEN_TRACKS': True,
    'CLEANUP_ARCHIVE': True,
    'ARCHIVE_CLEANUP_AGE': 9,
    'CLEANUP_EVENTS_TYPES': ['car'],
    # Add `safe_time_interval` in seconds to each `track` subject to delete
    # Prevents deletion of essential data.
    # Should be lower than CLEANUP_THRESHOLD
    'KEEP_EVENT_BEFORE_AFTER': 10, # minimal allowed value
    # Threshold between `intervals` in seconds.
    # If duration between `tracks` < `CLEANUP_THRESHOLD`
    # interval between tracks will not be added to deletion tasks
    'CLEANUP_THRESHOLD': 360, # minimal allowed value
},
```

4. Найдите параметр 'VMS_CLEANUP_SERVICE_SCHEDULE' и задайте рекуррентное правило (RRULE), по которому будет выполняться очистка архива.

Совет: См. калькулятор RRULE.

```
# rrule (recurrence rule) for scheduling `vms_cleanup` service
'VMS_CLEANUP_SERVICE_SCHEDULE': 'RRULE:FREQ=HOURLY;INTERVAL=3;WKST=MO',
```

5. Перезапустите контейнер findface-multi-findface-multi-legacy-1.

```
sudo docker container restart findface-multi-findface-multi-legacy-1
```

Ручная очистка видео

Вы можете вручную удалить видеофрагменты старше заданного количества дней, выполнив приведенную ниже команду. Для задания количества дней используйте аргумент `--vms-videos-max-age`.

```
sudo docker exec -it findface-multi-findface-multi-legacy-1 /opt/findface-security/bin/
↪python3 /tigre_prototype/manage.py cleanup_vms --vms-videos-max-age=30
```

2.4.12 Пользовательские вкладки, поля и фильтры в карточке

См.также:

Для создания пользовательских полей в базе данных векторов признаков см. *Пользовательские метаданные в Tarantool*.

Для добавления пользовательских вкладок и полей в карточки людей и транспортных средств выполните следующие действия:

1. Подготовьте список пользовательских вкладок и полей для добавления в карточки.
2. Откройте файл конфигурации `/opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py`.

```
sudo vi /opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py
```

3. Настройте карточки людей. Для этого раскомментируйте секцию `FFSECURITY -> CUSTOM_FIELDS -> human_card` и измените ее примерное содержимое с учетом следующих правил:

- 'items': список полей в карточке. Опишите каждое поле следующими параметрами:
 - 'name': внутреннее имя поля, string.
 - default: значение по умолчанию. Если значение по умолчанию больше `1e14 - 1`, то его следует записывать в виде строки, т. е. "123123..." вместо 123123...
 - 'label': название поля в карточке, string.
 - 'tab': вкладка, на которой располагается поле.
 - 'display': формат отображения (form или list), string или array.
 - 'description': описание поля, string.
 - 'editable': редактируемость поля, boolean.

- 'type': тип данных поля, string. Возможные значения:
 - * list: требует задания items, дополнительного параметра для списков (см. ниже), ожидает объекты {id, name} в словарях;
 - * valuelist: ожидает элементы примитивных типов.
 - * objectlist: позволяет создавать массивы объектов нужного типа.
 - * datetime: примитивный тип данных, отображаемый как список datetime.
 - * date: примитивный тип данных, отображаемый как выбор даты.
 - * boolean: примитивный тип данных, отображаемый как флажок.
 - * string: примитивный тип данных string.
- дополнительные параметры для списков (type=list, type=valuelist):
 - * multiple: возможность выбора нескольких элементов в списке, boolean.
 - * items: словарь, используемый как источник данных для списка.
 - * allow_create: возможность добавления новых элементов в список, boolean.
 - * custom_id: пользовательское поле для id (type=list).
- дополнительные параметры для списков объектов (type=objectlist).
 - * object: объекты, используемые как источник данных для списка объектов.
 - * simple: указывает, что поле ожидает данные примитивного типа вместо объектов, например, ожидает строки с телефонными номерами.
- 'filters': список фильтров для поиска по пользовательским полям. Параметры:
 - 'name': внутреннее имя фильтра,
 - 'label': название фильтра в веб-интерфейсе,
 - 'field': связанное поле в формате [имя поля].
- 'tabs': список полей в карточке.

```
FFSECURITY = {
...
# -- Custom model fields --
# Edit CUSTOM_FIELDS -> `human_card` section to customize human card fields.
# Edit CUSTOM_FIELDS -> `car_card` section to customize car card fields.
...
  'CUSTOM_FIELDS': {
    'human_card': {
      'items': [
        {
          'name': 'personid',
          'default': '',
          'label': 'PersonID',
          'display': ['list', 'form'],
          'description': 'Sigur person ID',
          'editable': False
        },

```

(continues on next page)

(продолжение с предыдущей страницы)

```

    {
      'name': 'firstname',
      'default': '',
      'label': 'First Name',
      'display': ['list', 'form'],
      'description': 'Sigur first name',
      'editable': False
    },
    {
      'name': 'lastname',
      'default': '',
      'label': 'Last Name',
      'display': ['list', 'form'],
      'description': 'Sigur last name',
      'editable': False
    },
    {
      'name': 'version',
      'default': '',
      'label': 'Version',
      'display': ['list', 'form'],
      'description': 'Sigur photo version',
      'editable': False
    }
  ],
  'filters': [
    {
      'name': 'personid',
      'label': 'Sigur person ID filter',
      'field': 'personid'
    }
  ]
},
'car_card': {}, # same fields are available
}

```

4. Настройте карточки транспортных средств. Для этого продублируйте содержимое секции `human_card` в секцию `car_card` и измените его по аналогии.
5. Перезапустите контейнер `findface-multi-findface-multi-legacy-1`.

```
sudo docker container restart findface-multi-findface-multi-legacy-1
```

Вы увидите, что в карточках появилось новое содержимое.

2.4.13 Пользовательские метаданные в Tarantool

Часто бывает необходимо присвоить дополнительные метаданные объектам, извлеченным из загруженных в картотеку изображений и теперь хранящимся в базе данных векторов признаков.

В этом разделе:

- *Настройка метаполей для лиц*
- *Настройка метаполей для силуэтов и транспортных средств*

Настройка метаполей для лиц

Для присвоения пользовательских метаданных лицам выполните следующие действия:

1. Подготовьте список пользовательских метаполей, которые будут присвоены лицам.
2. Откройте файл конфигурации `/opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py`.

```
sudo vi /opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py
```

3. В разделе `FFSECURITY` раскомментируйте секцию `CUSTOM_FIELDS -> face_object` и измените ее примерное содержимое с учетом следующих правил:
 - `field_name`: имя поля;
 - `type`: тип данных (`uint`, `string` или `bool`);
 - `default`: значение по умолчанию. Если значение по умолчанию больше `'1e14 - 1'`, то его следует записывать в виде строки, т. е. `"123123..."` вместо `123123...`

```
FFSECURITY = {
...
    # -- Custom model fields --
    ...
    # Edit CUSTOM_FIELDS -> `face_object` section to customize face object fields.
    ...
    # 'CUSTOM_FIELDS': {
        ...
        'face_object': {
            'items': [
                {
                    "field_name": "tag_name_1",
                    "type": "string",
                    "default": "change_me"
                },
                {
                    "field_name": "tag_name_2",
                    "type": "uint",
                    "default": 123
                }
            ]
        }
    }
}
```

(continues on next page)

(продолжение с предыдущей страницы)

```

        },
        {
            "field_name": "tag_name_3",
            "type": "bool",
            "default": True
        },
    ]
},
},
}

```

4. Добавьте новые метаполя в структуру базы данных векторов признаков.
5. Перезапустите контейнер `findface-multi-findface-multi-legacy-1`.

```
sudo docker container restart findface-multi-findface-multi-legacy-1
```

Вы можете работать с новыми метаполями через *HTTP API*, используя методы `objects/faces/`.

Настройка метаполей для силуэтов и транспортных средств

Назначение пользовательских метаполей силуэтам и транспортным средствам выполняется по аналогии с лицами. Единственное отличие состоит в том, что вам потребуется вручную создать разделы `CUSTOM_FIELDS -> body_object` и `CUSTOM_FIELDS -> car_object`. Рекомендуется продублировать раздел `CUSTOM_FIELDS -> face_object` и использовать дубликат в качестве отправной точки для дальнейших изменений.

```

FFSECURITY = {
...

    # -- Custom model fields --
    ...
    # Edit CUSTOM_FIELDS -> `face_object` section to customize face object fields.
    ...
    # 'CUSTOM_FIELDS': {
        ...
        'body_object': {
            'items': [
                {
                    "field_name": "tag_name_1",
                    "type": "string",
                    "default": "change_me"
                },
                {
                    "field_name": "tag_name_2",
                    "type": "uint",
                    "default": 123
                },
                {
                    "field_name": "tag_name_3",
                    "type": "bool",

```

(continues on next page)

(продолжение с предыдущей страницы)

```

        "default": True
    },
]
}
'car_object': {
    'items': [
        {
            "field_name": "tag_name_1",
            "type": "string",
            "default": "change_me"
        },
        {
            "field_name": "tag_name_2",
            "type": "uint",
            "default": 123
        },
        {
            "field_name": "tag_name_3",
            "type": "bool",
            "default": True
        },
    ],
}
},
}

```

Аналогично, вы можете работать с новыми метаполями, используя методы `objects/bodies/` и `objects/cars/ HTTP API`.

Примечание: Назначение пользовательских метаполей для событий лиц, силуэтов и транспортных средств выполняется по аналогии с объектами (лицами, силуэтам и транспортными средствами). Вам также потребуется вручную создать разделы `CUSTOM_FIELDS -> face_event`, `CUSTOM_FIELDS -> body_event` и `CUSTOM_FIELDS -> car_event` аналогичным образом.

См.также:

Для создания пользовательских вкладок, полей и фильтров в карточках см. *Пользовательские вкладки, поля и фильтры в карточке*.

2.4.14 Пакетная загрузка карточек через консоль

Помимо *веб-интерфейса* для пакетной загрузки фотографий в картотеку можно использовать консольную утилиту `uploader.py`. Рекомендуется предпочесть данную утилиту веб-интерфейсу, если количество загружаемых фотографий превышает 10 000.

Совет: Для вызова справки `uploader.py` выполните команду:

```
docker exec findface-multi-findface-multi-legacy-1 /opt/findface-security/bin/python3 /
↳ tigre_prototype/ffsecurity/uploader.py --help
```

(continues on next page)

(продолжение с предыдущей страницы)

Usage: uploader.py [OPTIONS] COMMAND [ARGS]...

Options:

```
--job FILE           Job file (default: enroll-job.db)
--log-level TEXT      Log level
--fsync BOOLEAN       Call fsync() to prevent data loss on power failure
--help               Show this message and exit.
```

Commands:

```
add      Add items from CSV or TSV file to job
print    Print contents of job file as JSON
run      Run upload job
```

```
docker exec findface-multi-findface-multi-legacy-1 /opt/findface-security/bin/python3 /
↳tigre_prototype/ffsecurity/uploader.py add --help
```

Usage: uploader.py add [OPTIONS] FILES...

Options:

```
--format [csv|tsv]   Input file format - CSV or TSV
--delimiter TEXT      Field delimiter - by default it's "\t" for TSV and ","
                      for CSV
--help               Show this message and exit.
```

```
docker exec findface-multi-findface-multi-legacy-1 /opt/findface-security/bin/python3 /
↳tigre_prototype/ffsecurity/uploader.py print --help
```

Usage: uploader.py print [OPTIONS]

Print contents of job file as JSON

Options:

```
--failed  Show only failed images
--noface  Show only images without detection
--help    Show this message and exit.
```

```
docker exec findface-multi-findface-multi-legacy-1 /opt/findface-security/bin/python3 /
↳tigre_prototype/ffsecurity/uploader.py run --help
```

Usage: uploader.py run [OPTIONS]

Run upload job

Options:

```
--parallel INTEGER    Number of enroll threads (default: 10)
--api TEXT             API url (default: http://127.0.0.1:80/)
                      [required]
--user TEXT            API username [required]
--password TEXT        API password [required]
```

(continues on next page)

(продолжение с предыдущей страницы)

```

--watch-lists TEXT      Comma-separated list of card list ids [required]
--inactive             Mark new cards as inactive
--failed              Include failed images
--noface              Include images without detection
--all-faces           Enroll all found faces on each image
--detect-timeout INTEGER Request timeout for detect photos
--logging-delta INTEGER Logging period delta
--help                Show this message and exit.

```

Выполните следующие действия:

1. Подготовьте CSV- или TSV-файл со списком фотографий и метаданными.

Важно: В качестве источника метаданных файл должен иметь следующий формат: путь к фотографии | метаданные.

Для подготовки TSV-файла можно использовать скрипт, аналогичный данному.

Примечание: Как скрипт, так и команда в примерах создают файл `images.tsv` с данными в формате полный путь к файлу с фотографией | метаданные. В качестве метаданных будет создана строка с именем файла.

Для запуска скрипта на создание TSV-файла со списком фотографий из домашней директории (`/home/ubuntu` в примере) выполните следующую команду:

```

sudo docker run -it --rm --network host --volume ${PWD}:/home/ubuntu/create_cards --
↪ volume /home/ubuntu/photos:/home/ubuntu/photos docker.int.ntl/ntech/multi/multi/
↪ legacy:ffmulti-2.0.0 sh -c "cd /home/ubuntu/create_cards && /opt/findface-
↪ security/bin/python3 tsv_builder.py /home/ubuntu/photos"

```

где `/home/ubuntu/photos` путь с вашим фотографиями.

2. Создайте файл задания (job-файл) из CSV- или TSV-файла, используя следующую команду. В результате в текущем каталоге будет создан файл `enroll-job.db`.

```

sudo docker run -it --rm --network host --volume ${PWD}:/home/ubuntu/create_cards_
↪ docker.int.ntl/ntech/multi/multi/legacy:ffmulti-2.0.0 sh -c "cd /home/ubuntu/
↪ create_cards && /opt/findface-security/bin/python3 /tigre_prototype/ffsecurity/
↪ uploader.py add /home/ubuntu/create_cards/images.tsv"

```

Опции команды `add` утилиты:

- `--format`: формат файла, по умолчанию `tsv`,
- `--delimiter`: используемый разделитель, по умолчанию `"\t"` для TSV-файла, `","` для формата CSV.

Примечание: Файл `job` представляет собой sqlite-базу, которая может быть открыта в консоли `sqlite3`.

3. Выполните задание job, указав путь к фотографиям вместо (например, /home/ubuntu/photos) и передав необходимые аргументы:

```
sudo docker run -it --rm --network host --volume ${PWD}:/home/ubuntu/create_cards --  
↪volume /home/ubuntu/photos:/home/ubuntu/photos docker.int.ntl/ntech/multi/multi/  
↪legacy:ffmulti-2.0.0 sh -c "cd /home/ubuntu/create_cards && /opt/findface-  
↪security/bin/python3 /tigre_prototype/ffsecurity/uploader.py run --user admin --  
↪password password --watch-lists 1"
```

Опции команды run утилиты:

- **--parallel**: количество потоков загрузки фотографий, по умолчанию 10. Чем больше потоков, тем быстрее будет завершена загрузка, однако также потребуется и большее количество ресурсов.
 - **--api**: API URL, компонента findface-security, по умолчанию http://127.0.0.1:80/. Обязательная опция.
 - **--user**: имя пользователя. Обязательная опция.
 - **--password**: пароль. Обязательная опция.
 - **--watch-lists**: перечень разделенных запятой id списков наблюдения, в которые нужно добавить фотографии. Обязательная опция.
 - **--inactive**: отметить новые карточки как неактивные.
 - **--failed**: в случае неудачи при обработке job-файла исправьте ошибку и повторите попытку с данной опцией.
 - **--noface**: по умолчанию изображения, классифицированные как не имеющие лиц, получают статус NOFACE и автоматически исключаются из загрузки. Чтобы попытаться повторно обнаружить лица на таких изображениях, перезапустите job-задание с данной опцией. Если повторное обнаружение снова даст отрицательный результат, изображение будет пропущено, а в журнале загрузки появится соответствующая запись.
 - **--all-faces**: загрузить все лица, если на фотографии их несколько.
 - **--detect-timeout**: тайм-аут запросов на обнаружение фотографий.
 - **--logging-delta**: частота логирования статуса загрузки.
4. (Опционально) Выведите в формате JSON результаты обработки задания job. При необходимости можно вывести только изображения, которые не удалось загрузить, и изображения без обнаруженных лиц.

Опции команды print утилиты:

- **--failed**: вывести только изображения, которые не удалось загрузить.
- **--noface**: вывести только изображения без обнаруженных лиц.

2.5 Настройка нейронных сетей FindFace Multi

2.5.1 Обзор моделей нейронных сетей

В этом разделе вы найдете сводную информацию по моделям нейронных сетей, созданным в нашей лаборатории и используемым в FindFace Multi.

Установленные модели расположены в каталоге `/opt/findface-multi/models/`.

Важно: При чистой установке биометрической моделью по умолчанию является `mango_320`.

Обнаружение лиц, транспортных средств, силуэтов

```
ls /opt/findface-multi/models/detector/

body.gustav_normal.015.cpu.fnk  body.jasmine_fast.018.cpu.fnk  car.gustav_accurate.004.
↪cpu.fnk  car.jasmine_fast.005.cpu.fnk  face.jasmine_fast.003.cpu.fnk  headbodyface.
↪alpha000_normal.001.gpu.fnk
body.gustav_normal.015.gpu.fnk  body.jasmine_fast.018.gpu.fnk  car.gustav_accurate.004.
↪gpu.fnk  car.jasmine_fast.005.gpu.fnk  face.jasmine_fast.003.gpu.fnk
```

Нормализация изображений лиц и силуэтов

```
ls /opt/findface-multi/models/facenorm/

bee.v3.cpu.fnk      bee_fast.gpu.fnk      crop2x.v2_maxsize400.cpu.fnk  crop2x.v2_
↪no_maxsize.gpu.fnk  facenorm.multicrop_full_center_size400.cpu.fnk  facenorm.multicrop_
↪full_crop2x_size400.gpu.fnk
bee.v3.gpu.fnk      crop1x.v2_maxsize400.cpu.fnk  crop2x.v2_maxsize400.gpu.fnk  cropbbox.
↪v2.cpu.fnk      facenorm.multicrop_full_center_size400.gpu.fnk
bee_fast.cpu.fnk  crop1x.v2_maxsize400.gpu.fnk  crop2x.v2_no_maxsize.cpu.fnk  cropbbox.
↪v2.gpu.fnk      facenorm.multicrop_full_crop2x_size400.cpu.fnk
```

Распознавание лиц

```
ls /opt/findface-multi/models/face/

lime.v2.cpu.fnk  lime.v2.gpu.fnk  mango_320.cpu.fnk  mango_320.gpu.fnk
```

Распознавание атрибутов лиц

```
ls /opt/findface-multi/models/faceattr/

age.v2.cpu.fnk      beard.v0.gpu.fnk      gender.v2.cpu.fnk      glasses3.v0.gpu.fnk  ↵
↪liveness.goodwin.cpu.fnk  liveness.pacs.v2.gpu.fnk  medmask3.v2.cpu.fnk      quality_
↪fast.v1.gpu.fnk
age.v2.gpu.fnk      emotions.v1.cpu.fnk    gender.v2.gpu.fnk      headpose.v2.cpu.fnk  ↵
↪liveness.goodwin.gpu.fnk  liveness.web.v0.cpu.fnk  medmask3.v2.gpu.fnk
beard.v0.cpu.fnk    emotions.v1.gpu.fnk    glasses3.v0.cpu.fnk    headpose.v2.gpu.fnk  ↵
↪liveness.pacs.v2.cpu.fnk  liveness.web.v0.gpu.fnk  quality_fast.v1.cpu.fnk
```

Нормализация изображений транспортных средств

```
ls /opt/findface-multi/models/carnorm/

anaferon.v5.cpu.fnk  anaferon.v5.gpu.fnk  anaferon.v7.cpu.fnk  anaferon.v7.gpu.fnk  ↵
↪briacon.v0.cpu.fnk  briacon.v0.gpu.fnk
```

Распознавание транспортных средств

```
ls /opt/findface-multi/models/carrec/

alonso.cpu.fnk  alonso.gpu.fnk
```

Распознавание атрибутов транспортных средств

```
ls /opt/findface-multi/models/carattr/

carattr.categories.v0.cpu.fnk      carattr.license_plate.v7.gpu.fnk      carattr.
↪orientation.v0.cpu.fnk  carattr.quality.v0.gpu.fnk      carattr.weight_types7.v0.
↪cpu.fnk  description.v0.gpu.fnk
carattr.categories.v0.gpu.fnk      carattr.license_plate_quality.v1.cpu.fnk  carattr.
↪orientation.v0.gpu.fnk  carattr.special_types11.v1.cpu.fnk  carattr.weight_types7.v0.
↪gpu.fnk
carattr.license_plate.v7.cpu.fnk  carattr.license_plate_quality.v1.gpu.fnk  carattr.
↪quality.v0.cpu.fnk      carattr.special_types11.v1.gpu.fnk  description.v0.cpu.fnk
```

Распознавание силуэтов

```
ls /opt/findface-multi/models/pedrec/

pedrec.clio.cpu.fnk  pedrec.clio.gpu.fnk
```

Распознавание атрибутов силуэтов

```
ls /opt/findface-multi/models/pedattr/

pedattr.age_gender.v0.cpu.fnk  pedattr.bags.v0.cpu.fnk  pedattr.clothes_type.v0.cpu.fnk  ↵
↵pedattr.color.v1.cpu.fnk  pedattr.protective.v1.cpu.fnk  pedattr.quality.v0.cpu.fnk
pedattr.age_gender.v0.gpu.fnk  pedattr.bags.v0.gpu.fnk  pedattr.clothes_type.v0.gpu.fnk  ↵
↵pedattr.color.v1.gpu.fnk  pedattr.protective.v1.gpu.fnk  pedattr.quality.v0.gpu.fnk
```

2.5.2 Распознавание лиц и их атрибутов

FindFace Multi позволяет распознавать человеческие лица и атрибуты лиц. Поддерживается распознавание таких атрибутов лица, как возраст, пол, эмоции, борода, очки, медицинские маски, положение головы, витальность.

Распознавание лиц и их атрибутов можно автоматически включить и настроить в ходе *установки FindFace Multi*. Если этот шаг был пропущен при установке, можно впоследствии выполнить настройку вручную. Распознавание лиц и атрибутов лиц работает как на GPU-, так и на CPU-ускорении.

Распознавание объекта лицо включено по умолчанию. Если во время установки FindFace Multi вы удалили распознавание объекта лицо, можно добавить его позже, следуя пошаговой инструкции ниже. Если распознавание лица уже установлено, и вам нужно только включить распознавание атрибутов лица, перейдите к шагам 1.5, 1.6 и 4.1, 4.2. Другие шаги следует пропустить.

1. Чтобы включить распознавание лиц, выполните следующее:

В файле конфигурации `/opt/findface-multi/configs/findface-extraction-api/findface-extraction-api.yaml` укажите модели нейронных сетей для распознавания объекта лицо и его атрибутов.

Важно: Удостоверьтесь, что для каждой модели вы указали правильный тип ускорения CPU или GPU: он должен совпадать с типом ускорения `findface-extraction-api`. Обратите внимание, что `findface-extraction-api` на CPU может работать только с CPU-моделями, в то время как `findface-extraction-api` на GPU поддерживает как GPU-, так и CPU-модели.

1. Откройте файл конфигурации `findface-extraction-api.yaml`.

```
sudo vi /opt/findface-multi/configs/findface-extraction-api/findface-extraction-
↵api.yaml
```

2. Укажите модель детектора лиц в разделе `detectors -> models`, вставив следующий код:

GPU

```
detectors:

...
models:
...
  face_jasmine:
    aliases:
      - face
```

(continues on next page)

(продолжение с предыдущей страницы)

```
- nnd
- cheetah
model: detector/face.jasmine_fast.003.gpu.fnk
options:
  min_object_size: 32
  resolutions:
    - 256x256
    - 384x384
    - 512x512
    - 768x768
    - 1024x1024
    - 1536x1536
    - 2048x2048

...
```

CPU

```
detectors:

...
models:
  ...
  face_jasmine:
    aliases:
      - face
      - nnd
      - cheetah
    model: detector/face.jasmine_fast.003.cpu.fnk
    options:
      min_object_size: 32
      resolutions:
        - 256x256
        - 384x384
        - 512x512
        - 768x768
        - 1024x1024
        - 1536x1536
        - 2048x2048

...
```

3. Убедитесь, что раздел `objects -> face` содержит `quality_attribute: face_quality` и, в зависимости от вашего типа ускорения, `base_normalizer: facenorm/crop2x.v2_maxsize400.gpu.fnk` или `base_normalizer: facenorm/crop2x.v2_maxsize400.cpu.fnk`:

GPU

```
objects:
  ...
  face:
    base_normalizer: facenorm/crop2x.v2_maxsize400.gpu.fnk
    quality_attribute: face_quality
  ...
```

CPU

```
objects:
  ...
  face:
    base_normalizer: facenorm/crop2x.v2_maxsize400.cpu.fnk
    quality_attribute: face_quality
  ...
```

4. В разделе `normalizers` укажите модели нормализаторов лица, вставив следующий код:

GPU

```
normalizers:
  ...
  models:
    crop1x:
      model: facenorm/crop1x.v2_maxsize400.gpu.fnk
    crop2x:
      model: facenorm/crop2x.v2_maxsize400.gpu.fnk
    cropbbox:
      model: facenorm/cropbbox.v2.gpu.fnk
    multicrop_full_center:
      model: facenorm/facenorm.multicrop_full_center_size400.gpu.fnk
    multicrop_full_crop2x:
      model: ''
    norm200:
      model: facenorm/bee.v3.gpu.fnk
  ...
```

CPU

```
normalizers:
  ...
  models:
    crop1x:
      model: facenorm/crop1x.v2_maxsize400.cpu.fnk
    crop2x:
      model: facenorm/crop2x.v2_maxsize400.cpu.fnk
    cropbbox:
```

(continues on next page)

(продолжение с предыдущей страницы)

```

    model: facenorm/cropbbox.v2.cpu.fnk
  multicrop_full_center:
    model: facenorm/facenorm.multicrop_full_center_size400.cpu.fnk
  multicrop_full_crop2x:
    model: ''
  norm200:
    model: facenorm/bee.v3.cpu.fnk
  ...

```

5.

Примечание: Необходимо выполнить этот шаг для включения распознавания атрибутов лица.

Для включения распознавания атрибутов лица выполните следующие действия:

В файле конфигурации `/opt/findface-multi/configs/findface-extraction-api/findface-extraction-api.yaml` укажите модели для экстракции векторов признаков в разделе `extractors`, как показано в примере ниже. Удостоверьтесь, что для каждой модели вы указали правильный тип ускорения CPU или GPU: он должен совпадать с типом ускорения `findface-extraction-api`.

GPU

```

extractors:
  ...
  models:
    face_age: faceattr/age.v2.gpu.fnk
    face_beard: faceattr/beard.v0.gpu.fnk
    face_beard4: ''
    face_countries47: ''
    face_emben: face/mango_320.gpu.fnk
    face_emotions: faceattr/emotions.v1.gpu.fnk
    face_eyes_attrs: ''
    face_eyes_openness: ''
    face_gender: faceattr/gender.v2.gpu.fnk
    face_glasses3: faceattr/glasses3.v0.gpu.fnk
    face_glasses4: ''
    face_hair: ''
    face_headpose: faceattr/headpose.v2.gpu.fnk
    face_headwear: ''
    face_highlight: ''
    face_liveness: faceattr/liveness.web.v0.gpu.fnk
    face_luminance_overexposure: ''
    face_luminance_underexposure: ''
    face_luminance_uniformity: ''
    face_medmask3: faceattr/medmask3.v2.gpu.fnk
    face_medmask4: ''
    face_mouth_attrs: ''
    face_quality: faceattr/quality_fast.v1.gpu.fnk
    face_scar: ''
    face_sharpness: ''

```

(continues on next page)

(продолжение с предыдущей страницы)

```
face_tattoo: ''
face_validity: ''
```

CPU

```
extractors:
    ...
models:
    face_age: faceattr/age.v2.cpu.fnk
    face_beard: faceattr/beard.v0.cpu.fnk
    face_beard4: ''
    face_countries47: ''
    face_emben: face/mango_320.cpu.fnk
    face_emotions: faceattr/emotions.v1.cpu.fnk
    face_eyes_attrs: ''
    face_eyes_openness: ''
    face_gender: faceattr/gender.v2.cpu.fnk
    face_glasses3: faceattr/glasses3.v0.cpu.fnk
    face_glasses4: ''
    face_hair: ''
    face_headpose: faceattr/headpose.v2.cpu.fnk
    face_headwear: ''
    face_highlight: ''
    face_liveness: faceattr/liveness.web.v0.cpu.fnk
    face_luminance_overexposure: ''
    face_luminance_underexposure: ''
    face_luminance_uniformity: ''
    face_medmask3: faceattr/medmask3.v2.cpu.fnk
    face_medmask4: ''
    face_mouth_attrs: ''
    face_quality: faceattr/quality_fast.v1.cpu.fnk
    face_scar: ''
    face_sharpness: ''
    face_tattoo: ''
    face_validity: ''
```

Доступны следующие модели экстракторов:

Экстрактор	Ускоре- ние	Настройка
возраст	CPU	face_age: faceattr/age.v2.cpu.fnk
	GPU	face_age: faceattr/age.v2.gpu.fnk
борода	CPU	face_beard: faceattr/beard.v0.cpu.fnk
	GPU	face_beard: faceattr/beard.v0.gpu.fnk
вектор признаков объекта лицо	CPU	face_emben: face/mango_320.cpu.fnk
	GPU	face_emben: face/mango_320.gpu.fnk
пол	CPU	face_gender: faceattr/gender.v2.cpu.fnk
	GPU	face_gender: faceattr/gender.v2.gpu.fnk
эмоции	CPU	face_emotions: faceattr/emotions.v1.cpu.fnk
	GPU	face_emotions: faceattr/emotions.v1.gpu.fnk
очки	CPU	face_glasses3: faceattr/glasses3.v0.cpu.fnk
	GPU	face_glasses3: faceattr/glasses3.v0.gpu.fnk
положение головы	CPU	face_headpose: faceattr/headpose.v2.cpu.fnk
	GPU	face_headpose: faceattr/headpose.v2.gpu.fnk
витальность лица	CPU	face_liveness: faceattr/liveness.web.v0.cpu.fnk
	GPU	face_liveness: faceattr/liveness.web.v0.gpu.fnk
медицинская маска	CPU	face_medmask3: faceattr/medmask3.v2.cpu.fnk
	GPU	face_medmask3: faceattr/medmask3.v2.gpu.fnk
качество изображения лица	CPU	face_quality: faceattr/quality_fast.v1.cpu.fnk
	GPU	face_quality: faceattr/quality_fast.v1.gpu.fnk

Совет: Для того чтобы отключить модель распознавания, передайте в соответствующий параметр пустое значение ''. Не удаляйте сам параметр, поскольку в этом случае будет выполняться поиск модели по умолчанию.

```
extractors:
  face_age: ''
  face_beard: ''
  face_beard4: ''
  face_countries47: ''
  face_emben: ''
  face_emotions: ''
  face_eyes_attrs: ''
  face_eyes_openness: ''
  face_gender: ''
  face_glasses3: ''
```

(continues on next page)

(продолжение с предыдущей страницы)

```

face_glasses4: ''
face_hair: ''
face_headpose: ''
face_headwear: ''
face_highlight: ''
face_liveness: ''
face_luminance_overexposure: ''
face_luminance_underexposure: ''
face_luminance_uniformity: ''
face_medmask3: ''
face_medmask4: ''
face_mouth_attrs: ''
face_quality: ''
face_scar: ''
face_sharpness: ''
face_tattoo: ''
face_validity: ''

```

Важно: Модель экстрактора для определения витальности лица `liveness.web.v0` включена по умолчанию. Не выключайте ее, если используется *аутентификация* по лицу пользователя.

6. Перезапустите контейнер `findface-multi-findface-extraction-api-1`.

```
sudo docker container restart findface-multi-findface-extraction-api-1
```

2. Чтобы включить распознавание лиц, внесите изменения в конфигурационный файл `/opt/findface-multi/configs/findface-video-worker/findface-video-worker.yaml`.

1. В разделе `models` укажите модели нейронных сетей для объекта `face` по аналогии с примером:

GPU

```

sudo vi /opt/findface-multi/configs/findface-video-worker/findface-video-worker.
↪yaml

models:
  ...
  detectors:
    ...
    face:
      fnk_path: /usr/share/findface-data/models/detector/face.jasmine_fast.003.
↪gpu.fnk
      min_size: 60
    ...
  normalizers:
    ...
    face_norm:
      fnk_path: /usr/share/findface-data/models/facenorm/crop2x.v2_maxsize400.

```

(continues on next page)

(продолжение с предыдущей страницы)

```

↪gpu.fnk
  face_norm_quality:
    fnk_path: /usr/share/findface-data/models/facenorm/crop1x.v2_maxsize400.
↪gpu.fnk
  ...
  extractors:
    ...
  face_quality:
    fnk_path: /usr/share/findface-data/models/faceattr/quality_fast.v1.gpu.fnk
    normalizer: face_norm_quality

```

CPU

```

sudo vi /opt/findface-multi/configs/findface-video-worker/findface-video-worker.
↪yaml

models:
  ...
  detectors:
    ...
    face:
      fnk_path: /usr/share/findface-data/models/detector/face.jasmine_fast.003.
↪cpu.fnk
      min_size: 60
    ...
  normalizers:
    ...
    face_norm:
      fnk_path: /usr/share/findface-data/models/facenorm/crop2x.v2_maxsize400.
↪cpu.fnk
      face_norm_quality:
        fnk_path: /usr/share/findface-data/models/facenorm/crop1x.v2_maxsize400.
↪cpu.fnk
    ...
  extractors:
    ...
    face_quality:
      fnk_path: /usr/share/findface-data/models/faceattr/quality_fast.v1.cpu.fnk
      normalizer: face_norm_quality

```

2. Убедитесь, что добавлен раздел `objects -> face`:

```

objects:
  ...
  face:
    normalizer: face_norm
    quality: face_quality
    track_features: ''

```

3. Перезапустите контейнер `findface-multi-findface-video-worker-1`.

```
sudo docker container restart findface-multi-findface-video-worker-1
```

- Для включения распознавания лиц, откройте файл конфигурации `/opt/findface-multi/configs/findface-video-manager/findface-video-manager.yaml` и убедитесь, что он содержит раздел `face` в `detectors`, аналогичный приведенному ниже.

```
sudo vi /opt/findface-multi/configs/findface-video-manager/findface-video-manager.
↪yaml

detectors:
  ...
  face:
    filter_min_quality: 0.5
    filter_min_size: 60
    filter_max_size: 8192
    roi: ''
    fullframe_crop_rot: false
    fullframe_use_png: false
    jpeg_quality: 95
    overall_only: true
    realtime_post_first_immediately: false
    realtime_post_interval: 1
    realtime_post_every_interval: false
    track_interpolate_bboxes: true
    track_miss_interval: 1
    track_overlap_threshold: 0.25
    track_max_duration_frames: 0
    track_send_history: false
    post_best_track_frame: true
    post_best_track_normalize: true
    post_first_track_frame: false
    post_last_track_frame: false
    tracker_type: simple_iou
    track_deep_sort_matching_threshold: 0.65
    track_deep_sort_filter_unconfirmed_tracks: true
    track_object_is_principal: false
    track_history_active_track_miss_interval: 0
```

- **Примечание:** Необходимо выполнить этот шаг для включения распознавания атрибутов лица.

Включите распознавание атрибутов лица в файле конфигурации `/opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py`.

- В разделе `FFSECURITY` укажите атрибуты лица, которые требуется отображать в событиях распознавания.

```
# available features: age, beard, emotions, gender, glasses, headpose, medmask
'FACE_EVENTS_FEATURES': ['glasses', 'beard', 'age', 'gender', 'headpose',
↪ 'medmask', 'emotions'],
```

- Перезапустите контейнер `findface-multi-findface-multi-legacy-1`.

```
sudo docker container restart findface-multi-findface-multi-legacy-1
```

2.5.3 Распознавание положения головы

Функциональность распознавания положения головы — это возможность камеры обнаруживать и отслеживать положение и движение головы человека относительно камеры видеонаблюдения в реальном времени.

Предупреждение: Функциональность распознавания положения головы не работает, если человек носит медицинскую маску.

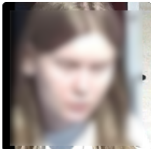
Для распознавания положения головы (поворот и наклон) FindFace Multi использует параметры pitch и yaw.

- Pitch — угол наклона головы вверх/вниз (т.е. относительно горизонтальной оси). Положительное значение pitch показывает, что голова наклонена вперед, а отрицательное указывает на наклон головы назад.
- Yaw — угол поворота головы вправо/влево (т.е. относительно вертикальной оси). Положительное значение yaw указывает на то, что голова повернута направо, а отрицательное — налево.

Углы pitch и yaw измеряются относительно камеры и принимают значения от -90 до +90 градусов.

Вы можете увидеть значения углов поворота и наклона головы в разделе «Атрибуты» для событий и фильтровать события лиц по этим параметрам.

Событие без совпадения ⋮ × ↘



Списки наблюдения ⋮

☐ Unmatched

Камеры ⋮

▶ ИТД, офис / Вход в главный корпус

Атрибуты ⋮

Женщина	Очки отсутствуют
21 год	Поворот: -21°
Нейтральное выражение	Наклон: -15°
Борода отсутствует	Маска отсутствует

Подтверждение ⋮

☒ Подтверждено 2023-05-25 15:13:43

Другое ⋮

ID 4523182597045116565

Создано 25.05.2023 15:13:43

Возможные сценарии использования распознавания положения головы

Функциональность распознавания положения головы можно использовать в сценариях для улучшения точности и повышения безопасности, например:

- Система контроля управления доступом (СКУД): улучшение системы контроля доступа сотрудников с помощью фиксации того, что лицо сотрудника совпадает с ожидаемым положением. Это значит, что если человек стоит рядом с камерой, поворачивает голову к камере, но не собирается проходить через СКУД, доступ не будет предоставлен благодаря установленному заранее значению положения головы.
- Аналитика посетителей ТЦ: уменьшение количества созданных некачественных кластеров по лицам. При кластеризации событий по лицам учитываются события соответствующие заданным значениям углов поворота головы. *Подробнее см. Кластеры.*

Как настроить функциональность распознавания положения головы

Чтобы настроить пороговое значение для формирования кластеров лиц, выполните следующее:

1. Откройте конфигурационный файл `/opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py`.

```
sudo vi /opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py
```

2. Включите пороговое значение положения головы, установив `'FACE_CLUSTER_EVENT_HEADPOSE_THRESHOLDS_ENABLE': True,`
3. При необходимости измените максимальный и минимальный порог для углов поворота и наклона.

```
'FACE_CLUSTER_EVENT_YAW_ANGLE_LOWEST_THRESHOLD': -30,
'FACE_CLUSTER_EVENT_YAW_ANGLE_HIGHEST_THRESHOLD': 30,
'FACE_CLUSTER_EVENT_PITCH_ANGLE_LOWEST_THRESHOLD': -60,
'FACE_CLUSTER_EVENT_PITCH_ANGLE_HIGHEST_THRESHOLD': 60,
```

4. Перезапустите контейнер `findface-multi-findface-multi-legacy-1`.

```
sudo docker container restart findface-multi-findface-multi-legacy-1
```

2.5.4 Распознавание транспортных средств и их атрибутов

FindFace Multi может распознавать отдельные транспортные средства и их атрибуты.

Атрибуты ТС следующие:

- номерной знак (для отдельных стран),

Примечание: Поддерживается распознавание номерных знаков для следующих стран:

Европа: Россия, Литва, Латвия, Эстония, Финляндия, Чехия, Сербия, Белоруссия, Украина, Молдавия, Грузия, Азербайджан, Армения.

Азия: ОАЭ, Казахстан, Кыргызстан, Таджикистан, Туркменистан, Узбекистан, Саудовская Аравия, Вьетнам, Индия, Пакистан, Таиланд.

Северная Америка: Мексика.

Южная Америка: Аргентина, Бразилия.

- цвет,
- марка,
- модель,
- тип кузова,
- категория ТС,
- ракурс (вид спереди, сзади или сбоку),
- принадлежность к *спецтранспорту* (такси, маршрутный транспорт, каршеринг, скорая помощь, полиция, спасательные службы, газоспасательные и аварийно-спасательные службы, военная техника, техника дорожных служб, остальной спецтранспорт),
- тип ТС по весу и размеру кузова.

Примечание: Распознавание спецтранспорта, так же как и распознавание ТС по весу и размеру кузова, работает только для некоторых стран. За дополнительной информацией обращайтесь к вашему менеджеру или в службу поддержки (support@ntechlab.com).

Распознавание транспортных средств вместе с выбором атрибутов распознавания можно настроить на уровне *установки* продукта. В этом разделе описывается как включить распознавание ТС и их атрибутов, если этот шаг был пропущен при установке.

Для включения распознавания транспортных средств и их атрибутов выполните следующие действия:

1. В файле конфигурации `/opt/findface-multi/configs/findface-extraction-api/findface-extraction-api.yaml` укажите модели нейронных сетей для распознавания транспортных средств и распознавания атрибутов транспортных средств.

Важно: Удостоверьтесь, что для каждой модели вы указали правильный тип ускорения: CPU или GPU. Он должен совпадать с типом ускорения `findface-extraction-api`. Обратите внимание, что `findface-extraction-api` на CPU может работать только с CPU-моделями, в то время как `findface-extraction-api` на GPU поддерживает как GPU-, так и CPU-модели.

1. Откройте файл конфигурации `findface-extraction-api.yaml`.

```
sudo vi /opt/findface-multi/configs/findface-extraction-api/findface-extraction-  
api.yaml
```

2. Укажите модель детектора ТС в разделе `detectors -> models`, вставив следующий код:

GPU

```

detectors:

...
models:
...
  gustav:
    aliases:
    - car
    - efreitor
    model: detector/car.gustav_accurate.004.gpu.fnk
    options:
      min_object_size: 32
      resolutions:
      - 256x256
      - 384x384
      - 512x512
      - 768x768
      - 1024x1024
      - 1536x1536
      - 2048x2048
...

```

CPU

```

detectors:

...
models:
...
  gustav:
    aliases:
    - car
    - efreitor
    model: detector/car.gustav_accurate.004.cpu.fnk
    options:
      min_object_size: 32
      resolutions:
      - 256x256
      - 384x384
      - 512x512
      - 768x768
      - 1024x1024
      - 1536x1536
      - 2048x2048
...

```

3. Убедитесь, что раздел `objects -> car` содержит `quality_attribute: car_quality` и, в зависимости от вашего типа ускорения, `base_normalizer: facenorm/cropbbox.v2.gpu.fnk` или `base_normalizer: facenorm/cropbbox.v2.cpu.fnk`.

GPU

```
objects:
  ...
  car:
    base_normalizer: facenorm/cropbbox.v2.gpu.fnk
    quality_attribute: car_quality
  ...
```

CPU

```
objects:
  ...
  car:
    base_normalizer: facenorm/cropbbox.v2.cpu.fnk
    quality_attribute: car_quality
  ...
```

4. Укажите нормализаторы, необходимые для работы экстракторов. Например, если вам требуется настроить распознавание автомобильных номеров, укажите нормализатор **carlicplate**. Для остальных экстракторов укажите нормализатор **cropbbox**.

Нормализатор	Модель нормализатора	Используется для экстракторов
carlicplate	carnorm/briacon.v0.gpu.fnk carnorm/briacon.v0.cpu.fnk	car_license_plate
cropbbox	facenorm/cropbbox.v2.gpu.fnk facenorm/cropbbox.v2.cpu.fnk	car_license_plate_quality, car_description, car_quality, car_special_types11, car_categories, car_orientation, car_weight_types7

GPU

```
normalizers:
  ...

models:
  ...
  carlicplate:
    model: carnorm/briacon.v0.gpu.fnk
  ...
  cropbbox:
    model: facenorm/cropbbox.v2.gpu.fnk
  ...
```

CPU

```
normalizers:
    ...

models:
    ...
    carlicplate:
        model: carnorm/briacon.v0.cpu.fnk
    ...
    cropbbox:
        model: facenorm/cropbbox.v2.cpu.fnk
    ...
```

5. Укажите модели необходимых экстракторов в разделе `extractors` -> `models`:

GPU

```
extractors:
    ...
models:
    car_categories: carattr/carattr.categories.v0.gpu.fnk
    car_color: ''
    car_container_number: ''
    car_description: carattr/description.v0.gpu.fnk
    car_emben: carrec/alonso.gpu.fnk
    car_license_plate: carattr/carattr.license_plate.v7.gpu.fnk
    car_license_plate_quality: carattr/carattr.license_plate_quality.v1.gpu.fnk
    car_license_plate_visibility: ''
    car_make: ''
    car_orientation: carattr/carattr.orientation.v0.gpu.fnk
    car_quality: carattr/carattr.quality.v0.gpu.fnk
    car_special_types: ''
    car_special_types11: carattr/carattr.special_types11.v1.gpu.fnk
    car_trash: ''
    car_weight_types7: carattr/carattr.weight_types7.v0.gpu.fnk
```

CPU

```
extractors:
    ...
models:
    car_categories: carattr/carattr.categories.v0.cpu.fnk
    car_color: ''
    car_container_number: ''
    car_description: carattr/description.v0.cpu.fnk
    car_emben: carrec/alonso.cpu.fnk
    car_license_plate: carattr/carattr.license_plate.v7.cpu.fnk
    car_license_plate_quality: carattr/carattr.license_plate_quality.v1.cpu.fnk
    car_license_plate_visibility: ''
```

(continues on next page)

(продолжение с предыдущей страницы)

```

car_make: ''
car_orientation: carattr/carattr.orientation.v0.cpu.fnk
car_quality: carattr/carattr.quality.v0.cpu.fnk
car_special_types: ''
car_special_types11: carattr/carattr.special_types11.v1.cpu.fnk
car_trash: ''
car_weight_types7: carattr/carattr.weight_types7.v0.cpu.fnk

```

Доступны следующие экстракторы:

Экстрактор	Настройка
вектор признаков от- дельного ТС	car_emben: carrec/alonso.cpu.fnk
	car_emben: carrec/alonso.gpu.fnk
номерной знак	car_license_plate: carattr/carattr.license_plate.v7. cpu.fnk car_license_plate_quality: carattr/carattr. license_plate_quality.v1.cpu.fnk
	car_license_plate: carattr/carattr.license_plate.v7. gpu.fnk car_license_plate_quality: carattr/carattr. license_plate_quality.v1.gpu.fnk
набор атрибутов: мар- ка / цвет / модель / тип кузова	car_description: carattr/description.v0.cpu.fnk
	car_description: carattr/description.v0.gpu.fnk
качество изображения ТС	car_quality: carattr/carattr.quality.v0.cpu.fnk
	car_quality: carattr/carattr.quality.v0.gpu.fnk
спецтранспорт	car_special_types: carattr/carattr.special_types11.v1. cpu.fnk
	car_special_types: carattr/carattr.special_types11.v1. gpu.fnk
категория ТС	car_categories: carattr/carattr.categories.v0.cpu.fnk
	car_categories: carattr/carattr.categories.v0.gpu.fnk
тип ТС по весу и раз- меру кузова	car_weight_types7: carattr/carattr.weight_types7.v0.cpu. fnk
	car_weight_types7: carattr/carattr.weight_types7.v0.gpu. fnk
ракурс ТС	car_orientation: carattr/carattr.orientation.v0.cpu.fnk
	car_orientation: carattr/carattr.orientation.v0.gpu.fnk

Совет: Для того чтобы отключить модель, передайте в соответствующий параметр пустое значение. Не удаляйте сам параметр, поскольку в этом случае будет выполняться поиск модели по умолчанию.

```

extractors:
...
models:
  car_categories: ''
  car_color: ''
  car_container_number: ''
  car_description: ''
  car_emben: ''

```

(continues on next page)

(продолжение с предыдущей страницы)

```

car_license_plate: ''
car_license_plate_quality: ''
car_license_plate_visibility: ''
car_make: ''
car_orientation: ''
car_quality: ''
car_special_types: ''
car_special_types11: ''
car_trash: ''
car_weight_types7: ''

```

6. Перезапустите контейнер `findface-multi-findface-extraction-api-1`.

```
sudo docker container restart findface-multi-findface-extraction-api-1
```

2. Внесите изменения в конфигурационный файл `/opt/findface-multi/configs/findface-video-worker/findface-video-worker.yaml`.

1. В разделе `models` укажите модели детектора, нормализатора и экстрактора для объекта `car`, как показано в примере ниже:

GPU

```

sudo vi /opt/findface-multi/configs/findface-video-worker/findface-video-worker.
↪yaml

models:
  ...
  detectors:
    ...
    car:
      fnk_path: /usr/share/findface-data/models/detector/car.jasmine_fast.005.
↪gpu.fnk
      min_size: 60
    ...
  normalizers:
    ...
    car_norm:
      fnk_path: /usr/share/findface-data/models/facenorm/cropbbox.v2.gpu.fnk
      car_norm_quality:
        fnk_path: /usr/share/findface-data/models/facenorm/cropbbox.v2.gpu.fnk
    ...
  extractors:
    ...
    car_quality:
      fnk_path: /usr/share/findface-data/models/carattr/carattr.quality.v0.gpu.
↪fnk
      normalizer: car_norm_quality

```

CPU

```

sudo vi /opt/findface-multi/configs/findface-video-worker/findface-video-worker.
↪yaml

models:
  ...
  detectors:
    ...
    car:
      fnk_path: /usr/share/findface-data/models/detector/car.jasmine_fast.005.
↪cpu.fnk
      min_size: 60
    ...
  normalizers:
    ...
    car_norm:
      fnk_path: /usr/share/findface-data/models/facenorm/cropbbox.v2.cpu.fnk
    car_norm_quality:
      fnk_path: /usr/share/findface-data/models/facenorm/cropbbox.v2.cpu.fnk
    ...
  extractors:
    ...
    car_quality:
      fnk_path: /usr/share/findface-data/models/carattr/carattr.quality.v0.cpu.
↪fnk
      normalizer: car_norm_quality

```

2. Убедитесь, что добавлен раздел `objects` -> `car`:

```

objects:
  ...
  car:
    normalizer: car_norm
    quality: car_quality
    track_features: ''

```

3. Перезапустите контейнер `findface-multi-findface-video-worker-1`.

```

sudo docker container restart findface-multi-findface-video-worker-1

```

3. Откройте файл конфигурации `/opt/findface-multi/configs/findface-video-manager/findface-video-manager.yaml` и убедитесь, что он содержит раздел `car` в `detectors`, аналогичный приведенному ниже. Обратите внимание, что для параметра `filter_min_quality` по умолчанию установлено значение `0.65`. Для получения более точных результатов с камер и других источников видео его можно увеличить. Рекомендованный порог равен `0.73`.

```

sudo vi /opt/findface-multi/configs/findface-video-manager/findface-video-manager.
↪yaml

detectors:
  ...

```

(continues on next page)

(продолжение с предыдущей страницы)

```

car:
  filter_min_quality: 0.65
  filter_min_size: 100
  filter_max_size: 8192
  roi: ''
  fullframe_crop_rot: false
  fullframe_use_png: false
  jpeg_quality: 95
  overall_only: true
  realtime_post_first_immediately: false
  realtime_post_interval: 1
  realtime_post_every_interval: false
  track_interpolate_bboxes: true
  track_miss_interval: 1
  track_overlap_threshold: 0.25
  track_max_duration_frames: 0
  track_send_history: false
  post_best_track_frame: true
  post_best_track_normalize: true
  post_first_track_frame: false
  post_last_track_frame: false
  tracker_type: simple_iou
  track_deep_sort_matching_threshold: 0.65
  track_deep_sort_filter_unconfirmed_tracks: true
  track_object_is_principal: false
  track_history_active_track_miss_interval: 0

```

Примечание: Настройка параметра `filter_min_quality` влияет только на результаты, получаемые от видеисточников, и не влияет на результаты подсчета транспортных средств и качество изображений транспортных средств, используемых в карточках. Чтобы получить более точные результаты подсчета или карточки ТС с изображениями лучшего качества, отдельно настройте параметр `MINIMUM_CAR_QUALITY` в файле конфигурации `findface-multi-legacy.py`. Эта настройка будет описана ниже.

4. Включите распознавание транспортных средств и атрибутов распознавания в файле конфигурации `/opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py`. Выполните следующие действия:

1. В разделе `FFSECURITY` установите `'ENABLE_CARS': True`.

```

sudo vi /opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.
→py

FFSECURITY = {
    ...
    # optional objects to detect
    'ENABLE_CARS': True,
    ...

```

2. В этом же разделе можно настроить параметр `MINIMUM_CAR_QUALITY`. По умолчанию для него установлено значение 0.65, что является рекомендуемым минимальным порогом качества изображения автомобиля, достаточным для создания карточки или для подсчета.

```
FFSECURITY = {
    ...
    # minimum car quality sufficient to add it to a card and use object in
    ↪ counter
    # p.s. set this parameter equals to `0.73` for more accurate results
    'MINIMUM_CAR_QUALITY': 0.65, # model: [carattr.quality.v0]
    ...
}
```

Если нужны более точные результаты подсчета или карточки автомобилей с изображениями лучшего качества, порог `MINIMUM_CAR_QUALITY` можно увеличить. Рекомендовано значение, равное 0.73.

3. В разделе `FFSECURITY` укажите атрибуты `TC`, которые требуется отображать в событиях распознавания.

```
# available features are: category, description, license_plate, orientation,
↪ special_vehicle_type, weight_type
'CAR_EVENTS_FEATURES': ['license_plate', 'category', 'special_vehicle_type',
↪ 'description', 'weight_type', 'orientation'],
```

4. Перезапустите контейнер `findface-multi-findface-multi-legacy-1`.

```
sudo docker container restart findface-multi-findface-multi-legacy-1
```

5. В веб-интерфейсе перейдите на вкладку *Источники видео*. Выберите камеру на вкладке *Камеры* (либо загруженный файл на вкладке *Загрузки*, либо внешний детектор на соответствующей вкладке). Перейдите на вкладку *Основные*. Поставьте флажок *TC* в разделе *Детекторы*.

2.5.5 Интерпретация результатов распознавания TC

Результаты распознавания транспортных средств можно увидеть в веб-интерфейсе на вкладке *Эпизоды и события*.

В некоторых случаях в результатах распознавания транспортных средств отсутствуют атрибуты описания и их значения, либо отображается, что значение атрибута неизвестно.

- Значение атрибута неизвестно. Если вы видите в результатах распознавания, что значение атрибута неизвестно, это означает, что достоверность распознавания атрибута ниже порога, указанного для атрибута в конфигурационном файле `/opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py`. В этом случае система не будет отображать распознанное значение. Оно будет утеряно.

Порог для атрибута транспортного средства можно настроить в файле конфигурации `findface-multi-legacy.py`.

Важно: Вышесказанное не относится к распознаванию регистрационных знаков (ГРЗ) транспортных средств. По умолчанию порог распознавания регистрационных знаков установлен на минимальное значение. Если вы видите в результатах распознавания, что регистрационный знак неизвестен, значит, на объекте он плохо различим или его совсем не видно. Распознавание страны, региона и цвета ГРЗ зависит от результатов распознавания регистрационного знака. Если ГРЗ не распознан, то и значения этих атрибутов будут отображаться как неизвестные.

Примечание: Регион и цвет регистрационного знака определяются только для Объединенных

Арабских Эмиратов (ОАЭ). Для других стран значения этих атрибутов будут выводиться как неизвестные в результатах распознавания.

- Отсутствие атрибутов описания и их значений. Если в результатах распознавания нет атрибутов описания (т.е. марки автомобиля, типа кузова, модели, цвета) и их значений, это происходит по следующим причинам:
 - Распознавание категорий транспортных средств, а также распознавание ТС по весу и размеру кузова все еще находится в стадии разработки. FindFace Multi уверенно распознает автомобили категории В и их атрибуты. Что касается распознавания ТС по весу и размеру кузова, поддерживается распознавание легковых и грузовых автомобилей до 3.5 тонн и их атрибутов. Однако система может давать ложноположительные результаты по другим категориям и весовым классам транспортных средств и их атрибутам описания. Поэтому атрибуты описания и значения их распознавания отображаются только для легковых автомобилей, а если говорить о весовых категориях — только для легковых и грузовых автомобилей до 3.5 тонн. Для других категорий и весовых классов ТС атрибуты описания и результаты их распознавания скрыты. Достоверность распознавания атрибутов описания в данном случае не имеет значения: она может быть ниже или выше порога.
 - То же самое происходит, если категория автомобиля неизвестна: атрибуты описания вместе со значениями распознавания не отображаются. Достоверность распознавания атрибутов описания в данном случае не является определяющим фактором.

При необходимости эту настройку можно изменить в разделе `CAR_EVENTS_FEATURES_TO_NULL` файла конфигурации `findface-multi-legacy.py`.

Читайте далее, чтобы узнать, как настроить пороговые значения для атрибутов ТС или включить отображение атрибутов описания для некоторых или всех категорий транспорта и неизвестных ТС.

В этом разделе:

- *Настройка порога для атрибутов ТС*
- *Настройка отображения атрибутов описания в результатах распознавания*

Настройка порога для атрибутов ТС

Для каждого атрибута транспортного средства по умолчанию выставлено оптимальное значение порога. При необходимости его можно изменить.

Откройте файл конфигурации `/opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py`. В разделе `FFSECURITY` измените пороговые значения для выбранных атрибутов:

```
sudo vi /opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py

FFSECURITY = {
    ...

    'CAR_DESCRIPTION_THRESHOLD': {
        'make': 0.5,
        'model': 0.5,
```

(continues on next page)

(продолжение с предыдущей страницы)

```

    'body': 0.5,
    'color': 0.5,
  }, # model: [description.v0]
  'SPECIAL_VEHICLE_TYPE_THRESHOLD': 0.5, # model: [special_types11.v1]
  'CAR_CATEGORY_THRESHOLD': 0.6, # model: [categories.v0]
  'CAR_WEIGHT_TYPE_THRESHOLD': 0.5, # model: [weight-types7.v0]
  'CAR_ORIENTATION_THRESHOLD': 0.8, # model [orientation.v0]
  ...

```

Перезапустите контейнер `findface-multi-findface-multi-legacy-1`.

```
sudo docker container restart findface-multi-findface-multi-legacy-1
```

Настройка отображения атрибутов описания в результатах распознавания

При необходимости можно включить отображение атрибутов описания для неподдерживаемых категорий транспортных средств или неизвестных транспортных средств. По умолчанию вывод атрибутов описания включен только для автомобилей категории В и для ТС весовых категорий В_light (легковой автомобиль массой до 3.5 тонн) и В_heavy (грузовой автомобиль массой до 3.5 тонн). Используйте информацию из таблиц, чтобы сопоставить транспортное средство и его категорию или класс в зависимости от веса.

Таблица 1: Категории транспортных средств

Категория	Транспортное средство
A	мотоцикл (включая мопед), скутер, квадроцикл
B	легковой автомобиль
BE	легковой автомобиль с прицепом
C	грузовик
CE	грузовик с прицепом
D	автобус
DE	сочлененный автобус
other	ТС, которое не относится ни к одной из вышеперечисленных категорий

Таблица 2: Весовые категории транспортных средств

Весовая категория	Транспортное средство
B_light	легковой автомобиль (до 3.5 т)
B_heavy	грузовик (до 3.5 т)
BE	легковой автомобиль с прицепом
C_light	грузовик (до 12 т)
C_heavy	тяжелый грузовик (свыше 12 т)
D_light	одиночный автобус
D_long	сочлененный автобус
other	ТС, которое не относится ни к одной из вышеперечисленных категорий

Предупреждение: Не рекомендуется менять значения, установленные по умолчанию. Прежде чем вносить какие-либо изменения, свяжитесь с нашими техническими специалистами (support@ntechlab.com).

Чтобы включить отображение атрибутов описания для неподдерживаемых категорий транспортных средств или для неизвестных ТС, откройте файл конфигурации `/opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py`. В разделе `FFSECURITY` найдите настройку `CAR_EVENTS_FEATURES_TO_NULL`:

```
sudo vi /opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py

FFSECURITY = {
    ...

    'CAR_EVENTS_FEATURES_TO_NULL': {
        'category': {
            # features, that will be nulled for all categories
            'default': ['make', 'body', 'model', 'color'],
            'unknown': ['make', 'body', 'model', 'color'],
            # categories with other behavior
            'B': [],
        },
        'weight_type': {
            # features, that will be nulled for all weight types
            'default': ['make', 'body', 'model', 'color'],
            # weight types with other behavior
            'B_light': [],
            'B_heavy': [],
        },
    },
    ...
}
```

Чтобы включить отображение атрибутов описания, удалите их из выбранного параметра. Например, если вы хотите включить распознавание атрибутов описания для всех категорий транспортных средств, кроме неизвестных транспортных средств, удалите их из параметра `default`. Обратите внимание, что неизвестные транспортные средства исключены из параметра `default` и настраиваются через параметр `unknown`.

```
sudo vi /opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py

FFSECURITY = {
    ...

    'CAR_EVENTS_FEATURES_TO_NULL': {
        'category': {
            # features, that will be nulled for all categories
            'default': [],
            'unknown': ['make', 'body', 'model', 'color'],
            # categories with other behavior
            'B': [],
        },
        'weight_type': {
```

(continues on next page)

(продолжение с предыдущей страницы)

```

        # features, that will be nulled for all weight types
        'default': [],
        # weight types with other behavior
        'B_light': [],
        'B_heavy': [],
    },
},
...

```

Чтобы настроить вывод атрибутов описания для определенной категории транспортного средства или для транспортного средства определенной весовой категории, добавьте название категории или весовой категории в исключения в разделах **categories with other behavior** или **weight types with other behavior**. Используйте информацию из таблиц *Категории транспортных средств* и *Весовые категории транспортных средств*, чтобы сопоставить ТС и его категорию или класс по весу. Например, если вы хотите добавить в исключение автобусы (категория D) и одиночные автобусы (весовая категория D_light), сделайте следующее:

```

sudo vi /opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py

FFSECURITY = {
    ...

    'CAR_EVENTS_FEATURES_TO_NULL': {
        'category': {
            # features, that will be nulled for all categories
            'default': ['make', 'body', 'model', 'color'],
            'unknown': ['make', 'body', 'model', 'color'],
            # categories with other behavior
            'B': [],
            'D': [],
        },
        'weight_type': {
            # features, that will be nulled for all weight types
            'default': ['make', 'body', 'model', 'color'],
            # weight types with other behavior
            'B_light': [],
            'B_heavy': [],
            'D_light': [],
        },
    },
},
...

```

Перезапустите контейнер findface-multi-findface-multi-legacy-1.

```
sudo docker container restart findface-multi-findface-multi-legacy-1
```

2.5.6 Распознавание спецтранспорта

FindFace Multi поддерживает распознавание спецтранспорта и определяет его тип. Распознавание спецтранспорта можно включить на этапе *установки* продукта или выполнить настройку позже, следуя инструкции в разделе *Распознавание транспортных средств и их атрибутов*.

Примечание: Распознавание спецтранспорта работает только для некоторых стран. За дополнительной информацией обращайтесь к вашему менеджеру или в службу поддержки (support@ntechlab.com).

Поддерживается распознавание следующих типов транспортных средств:

- Такси: такси, в том числе грузовое. В некоторых случаях грузовое такси может определяться как «остальной спецтранспорт».
- Общественный маршрутный транспорт: автобусы, маршрутки, троллейбусы.
- Каршеринг: поддерживаются следующие сервисы каршеринга: Ситидрайв, Яндекс.Драйв, Делимобиль, BelkaCar. Также могут распознаваться автомобили других операторов в случае сильной схожести с перечисленными выше.
- Скорая помощь и автомобили реанимации (белые и желтые автомобили).
- Полиция, в том числе автомобили ДПС.
- Спасательные службы: автомобили МЧС, в том числе пожарные машины.
- Газоспасательные и аварийно-спасательные службы: жёлтые и белые автомобили, в том числе с контрастными красными дверьми, характерными красными полосками и обозначением 04 / 104 на борту.
- Военная техника, в том числе автомобили Росгвардии.
- Техника дорожных служб: коммунальная (дорожная, поливомоечная и уборочная техника), строительная техника, автомобили ЦОДД.
- Остальной спецтранспорт: спецтехника, которая не относится к вышеперечисленным типам.
- Не спецтранспорт.

2.5.7 Распознавание силуэтов и их атрибутов

FindFace Multi позволяет распознавать отдельные человеческие силуэты и атрибуты силуэтов.

Атрибуты силуэтов следующие:

- пол:
 - мужской;
 - женский;
- возраст (по группам):
 - 0-16 лет;
 - 17-35 лет;
 - 36-50 лет;
 - 50+ лет;
- тип одежды:

- обобщенная категория одежды верхней части тела: с длинными рукавами, с короткими рукавами, без рукавов;
- детализированный тип одежды верхней части тела: куртка, пальто, жилет без рукавов, толстовка, футболка, рубашка, платье;
- тип одежды нижней части тела: брюки, юбка, шорты, неопределенный;
- тип головного убора: шапка/шляпа/кепка, капюшон/платок, без головного убора;
- цвет одежды (верх/низ);
- наличие средств индивидуальной защиты (СИЗ):
 - наименование СИЗ: жилет, каска;
 - цвет СИЗ;
 - точность распознавания СИЗ;
- наличие сумки:
 - на спине;
 - в руке(-ах).

Распознавание силуэтов и атрибутов силуэтов можно настроить на уровне *установки* продукта. В этом разделе описывается как включить распознавание силуэтов и атрибутов силуэтов, если этот шаг был пропущен при установке.

Для включения распознавания силуэтов и их атрибутов выполните следующие действия:

1. В файле конфигурации `/opt/findface-multi/configs/findface-extraction-api/findface-extraction-api.yaml` укажите модели нейронных сетей для распознавания объектов типа силуэт и их атрибутов.

Важно: Удостоверьтесь, что для каждой модели вы указали правильный тип ускорения CPU или GPU: он должен совпадать с типом ускорения `findface-extraction-api`. Обратите внимание, что `findface-extraction-api` на CPU может работать только с CPU-моделями, в то время как `findface-extraction-api` на GPU поддерживает как GPU-, так и CPU-модели.

1. Откройте файл конфигурации `findface-extraction-api.yaml`.

```
sudo vi /opt/findface-multi/configs/findface-extraction-api/findface-extraction-  
api.yaml
```

2. Укажите модель детектора силуэтов в разделе `detectors -> models`, вставив следующий код:

GPU

```
detectors:  
  
  ...  
  models:  
    ...  
    body_gustav:  
      aliases:
```

(continues on next page)

(продолжение с предыдущей страницы)

```

- body
- edie
- shiloette
- glen
model: detector/body.gustav_normal.015.gpu.fnk
options:
  min_object_size: 32
  resolutions:
    - 256x256
    - 384x384
    - 512x512
    - 768x768
    - 1024x1024
    - 1536x1536
    - 2048x2048
...

```

CPU

```

detectors:
...
models:
...
  body_gustav:
    aliases:
      - body
      - edie
      - shiloette
      - glen
    model: detector/body.gustav_normal.015.cpu.fnk
    options:
      min_object_size: 32
      resolutions:
        - 256x256
        - 384x384
        - 512x512
        - 768x768
        - 1024x1024
        - 1536x1536
        - 2048x2048
...

```

3. Убедитесь, что раздел `objects -> body` содержит `quality_attribute: body_quality` и, в зависимости от вашего типа ускорения, `base_normalizer: facenorm/croppbox.v2.gpu.fnk` или `base_normalizer: facenorm/croppbox.v2.cpu.fnk`:

GPU

```
objects:
  ...
  body:
    base_normalizer: facenorm/cropbbox.v2.gpu.fnk
    quality_attribute: body_quality
  ...
```

CPU

```
objects:
  ...
  body:
    base_normalizer: facenorm/cropbbox.v2.cpu.fnk
    quality_attribute: body_quality
  ...
```

4. Убедитесь, что раздел **normalizers** содержит модель нормализатора **cropbbox**, как показано в примере ниже. Данный нормализатор требуется для работы экстракторов.

GPU

```
normalizers:
  ...

models:
  ...
  cropbbox:
    model: facenorm/cropbbox.v2.gpu.fnk
  ...
```

CPU

```
normalizers:
  ...

models:
  ...
  cropbbox:
    model: facenorm/cropbbox.v2.cpu.fnk
  ...
```

5. Укажите модели для экстракции векторов признаков в разделе **extractors -> models**, в зависимости от необходимых экстракторов:

GPU

```
extractors:
    ...
    models:
        body_action_base6: ''
        body_action_car: ''
        body_action_fights: ''
        body_age_gender: pedattr/pedattr.age_gender.v0.gpu.fnk
        body_bags: pedattr/pedattr.bags.v0.gpu.fnk
        body_clothes: pedattr/pedattr.clothes_type.v0.gpu.fnk
        body_clothes34671: ''
        body_color: pedattr/pedattr.color.v1.gpu.fnk
        body_emben: pedrec/pedrec.clio.gpu.fnk
        body_fall: ''
        body_handface: ''
        body_protective_equipment: pedattr/pedattr.protective.v1.gpu.fnk
        body_quality: pedattr/pedattr.quality.v0.gpu.fnk
```

CPU

```
extractors:
    ...
    models:
        body_action_base6: ''
        body_action_car: ''
        body_action_fights: ''
        body_age_gender: pedattr/pedattr.age_gender.v0.cpu.fnk
        body_bags: pedattr/pedattr.bags.v0.cpu.fnk
        body_clothes: pedattr/pedattr.clothes_type.v0.cpu.fnk
        body_clothes34671: ''
        body_color: pedattr/pedattr.color.v1.cpu.fnk
        body_emben: pedrec/pedrec.clio.cpu.fnk
        body_fall: ''
        body_handface: ''
        body_protective_equipment: pedattr/pedattr.protective.v1.cpu.fnk
        body_quality: pedattr/pedattr.quality.v0.cpu.fnk
```

Доступны следующие экстракторы:

Экстрактор	Настройка
возраст и пол	body_age_gender: pedattr/pedattr.age_gender.v0.gpu.fnk
	body_age_gender: pedattr/pedattr.age_gender.v0.cpu.fnk
наличие сумки	body_bags: pedattr/pedattr.bags.v0.gpu.fnk
	body_bags: pedattr/pedattr.bags.v0.cpu.fnk
тип одежды	body_clothes: pedattr/pedattr.clothes_type.v0.gpu.fnk
	body_clothes: pedattr/pedattr.clothes_type.v0.cpu.fnk
цвет одежды	body_color: pedattr/pedattr.color.v1.gpu.fnk
	body_color: pedattr/pedattr.color.v1.cpu.fnk
вектор признаков объекта силуэт	body_emben: pedrec/pedrec.clio.gpu.fnk
	body_emben: pedrec/pedrec.clio.cpu.fnk
наличие СИЗ	body_protective_equipment: pedattr/pedattr. protective.v1.gpu.fnk
	body_protective_equipment: pedattr/pedattr. protective.v1.cpu.fnk
качество изображения си- луэта	body_quality: pedattr/pedattr.quality.v0.gpu.fnk
	body_quality: pedattr/pedattr.quality.v0.cpu.fnk

Совет: Для того чтобы отключить модель, передайте в соответствующий параметр пустое значение ''. Не удаляйте сам параметр, поскольку в этом случае будет выполняться поиск модели по умолчанию.

Важно: Для того чтобы распознавание силуэтов работало корректно, для параметров body_emben и body_quality должны быть указаны соответствующие модели экстракторов.

GPU

```
extractors:
...
models:
  body_action_base6: ''
  body_action_car: ''
  body_action_fights: ''
  body_age_gender: ''
  body_bags: ''
  body_clothes: ''
  body_clothes34671: ''
  body_color: ''
  body_emben: pedrec/pedrec.clio.gpu.fnk
  body_fall: ''
  body_handface: ''
  body_protective_equipment: ''
  body_quality: pedattr/pedattr.quality.v0.gpu.fnk
```

CPU

```
extractors:
  ...
models:
  body_action_base6: ''
  body_action_car: ''
  body_action_fights: ''
  body_age_gender: ''
  body_bags: ''
  body_clothes: ''
  body_clothes34671: ''
  body_color: ''
  body_emben: pedrec/pedrec.clio.cpu.fnk
  body_fall: ''
  body_handface: ''
  body_protective_equipment: ''
  body_quality: pedattr/pedattr.quality.v0.cpu.fnk
```

6. Перезапустите контейнер `findface-multi-findface-extraction-api-1`.

```
sudo docker container restart findface-multi-findface-extraction-api-1
```

2. Внесите изменения в конфигурационный файл `/opt/findface-multi/configs/findface-video-worker/findface-video-worker.yaml`.

1. В разделе `models` укажите модели нейронных сетей для объекта `body` по аналогии с примером:

GPU

```
sudo vi /opt/findface-multi/configs/findface-video-worker/findface-video-worker.
↪yaml

models:
  ...
  detectors:
    ...
    body:
      fnk_path: /usr/share/findface-data/models/detector/body.jasmine_fast.018.
↪gpu.fnk
      min_size: 60
    ...
  normalizers:
    ...
    body_norm:
      fnk_path: /usr/share/findface-data/models/facenorm/cropbbox.v2.gpu.fnk
    body_norm_quality:
      fnk_path: /usr/share/findface-data/models/facenorm/cropbbox.v2.gpu.fnk
    ...
  extractors:
    ...
```

(continues on next page)

(продолжение с предыдущей страницы)

```

    body_quality:
      fnk_path: /usr/share/findface-data/models/pedattr/pedattr.quality.v0.gpu.
↪fnk
      normalizer: body_norm_quality

```

CPU

```

sudo vi /opt/findface-multi/configs/findface-video-worker/findface-video-worker.
↪yaml

models:
  ...
  detectors:
    ...
    body:
      fnk_path: /usr/share/findface-data/models/detector/body.jasmine_fast.018.
↪cpu.fnk
      min_size: 60
    ...
  normalizers:
    ...
    body_norm:
      fnk_path: /usr/share/findface-data/models/facenorm/cropbbox.v2.cpu.fnk
    body_norm_quality:
      fnk_path: /usr/share/findface-data/models/facenorm/cropbbox.v2.cpu.fnk
    ...
  extractors:
    ...
    body_quality:
      fnk_path: /usr/share/findface-data/models/pedattr/pedattr.quality.v0.cpu.
↪fnk
      normalizer: body_norm_quality

```

2. Убедитесь, что добавлен раздел `objects` -> `body`:

```

objects:
  ...
  body:
    normalizer: body_norm
    quality: body_quality
    track_features: ''

```

3. Перезапустите контейнер `findface-multi-findface-video-worker-1`.

```

sudo docker container restart findface-multi-findface-video-worker-1

```

3. Откройте файл конфигурации `/opt/findface-multi/configs/findface-video-manager/findface-video-manager.yaml` и убедитесь, что он содержит раздел `body` в `detectors`, аналогичный приведенному ниже.

```

sudo vi /opt/findface-multi/configs/findface-video-manager/findface-video-manager.
↪yaml

detectors:
  ...
  body:
    filter_min_quality: 0.6
    filter_min_size: 70
    filter_max_size: 8192
    roi: ''
    fullframe_crop_rot: false
    fullframe_use_png: false
    jpeg_quality: 95
    overall_only: true
    realtime_post_first_immediately: false
    realtime_post_interval: 1
    realtime_post_every_interval: false
    track_interpolate_bboxes: true
    track_miss_interval: 1
    track_overlap_threshold: 0.25
    track_max_duration_frames: 0
    track_send_history: false
    post_best_track_frame: true
    post_best_track_normalize: true
    post_first_track_frame: false
    post_last_track_frame: false
    tracker_type: simple_iou
    track_deep_sort_matching_threshold: 0.65
    track_deep_sort_filter_unconfirmed_tracks: true
    track_object_is_principal: false
    track_history_active_track_miss_interval: 0

```

4. Включите распознавание силуэтов и атрибутов силуэтов в файле конфигурации `/opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py`. Выполните следующие действия:

1. В разделе FFSECURITY установите 'ENABLE_BODIES: True.

```

sudo vi /opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.
↪py

FFSECURITY = {
  ...

  # optional objects to detect
  'ENABLE_BODIES': True,
  ...
}

```

2. Чтобы улучшить качество распознавания силуэтов, мы рекомендуем включить дополнительный анализ атрибутов. В этом случае система сравнивает не только векторы признаков двух силуэтов, но и их атрибуты. Вывод о совпадении силуэтов делается только в том случае, если совпадают как векторы признаков, так и атрибуты.

Для дополнительного анализа вы можете использовать следующие атрибуты:

- `bottom_color`: цвет одежды нижней части тела;
- `top_color`: цвет одежды верхней части тела;
- `headwear`: тип и наличие/отсутствие головного убора;
- `detailed_upper_clothes`: детализированный тип одежды верхней части тела, например, куртка;
- `upper_clothes`: обобщенная категория одежды верхней части тела: с длинными рукавами, с короткими рукавами, без рукавов;
- `lower_clothes`: тип одежды нижней части тела, например, брюки;
- `bag_hand`: наличие сумки в руке(-ах);
- `bag_back`: наличие сумки на спине;
- `helmet_type`: включает такие характеристики, как цвет каски, определение ее видимости на человеке, отсутствие/наличие;
- `vest_type`: включает такие характеристики, как цвет жилета, определение его видимости на человеке, отсутствие/наличие;
- `age_group`: принадлежность к одной из четырех возрастных групп: 0-16, 17-35, 36-50, 50+ лет;
- `gender`: пол: мужской, женский.

Для того чтобы включить дополнительный анализ атрибутов, установите `True` в разделе `FFSECURITY -> EXTRA_BODY_MATCHING` для атрибутов, которые вы хотите сравнивать. Установите значение `min_confidence` в диапазоне от 0 до 1.

```
FFSECURITY = {  
    # use additional features for extra confidence when matching body by emben  
    'EXTRA_BODY_MATCHING': {  
        'bottom_color': {'enabled': False, 'min_confidence': 0},  
        'top_color': {'enabled': False, 'min_confidence': 0},  
        'headwear': {'enabled': False, 'min_confidence': 0},  
        'detailed_upper_clothes': {'enabled': False, 'min_confidence': 0},  
        'upper_clothes': {'enabled': False, 'min_confidence': 0},  
        'lower_clothes': {'enabled': False, 'min_confidence': 0},  
        'bag_hand': {'enabled': False, 'min_confidence': 0},  
        'bag_back': {'enabled': False, 'min_confidence': 0},  
        'bag_ground': {'enabled': False, 'min_confidence': 0},  
        'helmet_type': {'enabled': False, 'min_confidence': 0},  
        'vest_type': {'enabled': False, 'min_confidence': 0},  
        'age_group': {'enabled': False, 'min_confidence': 0},  
        'gender': {'enabled': False, 'min_confidence': 0},  
    },  
}
```

Примечание: Атрибут `bag_ground` пока недоступен для дополнительного анализа.

Примечание: Проконсультируйтесь с нашими техническими экспертами (support@ntechlab.com), чтобы установить оптимальное значение для `min_confidence`.

Если вы решили, что дополнительный анализ атрибутов вам не нужен, пропустите эту настройку и переходите к следующему шагу.

- В разделе **FFSECURITY** укажите атрибуты силуэтов, которые требуется отображать в событиях распознавания.

```
# available features: age_gender, bags, clothes, color, protective_equipment
'BODY_EVENTS_FEATURES': ['protective_equipment', 'age_gender', 'bags', 'color',
↪ 'clothes'],
```

- Перезапустите контейнер `findface-multi-findface-multi-legacy-1`.

```
sudo docker container restart findface-multi-findface-multi-legacy-1
```

- В веб-интерфейсе перейдите на вкладку *Источники видео*. Выберите камеру на вкладке *Камеры* (либо загруженный файл на вкладке *Загрузки*, либо внешний детектор на соответствующей вкладке). Перейдите на вкладку *Основные*. Поставьте флажок *Силуэты* в разделе Детекторы.

2.5.8 Включение определения витальности лиц

Детектор витальности лиц FindFace Multi отличает живые лица от изображений, таких как фото, видеозапись или маска. Детектор оценивает витальность лица с определенным уровнем достоверности и возвращает оценку достоверности вместе с бинарным результатом **истинное лицо/изображение**, в зависимости от предустановленного порога достоверности.

Детектор витальности лиц можно автоматически включить и настроить в ходе *установки* продукта. При пропуске данного шага можно впоследствии сделать это вручную, руководствуясь описанием в данном разделе.

Примечание: Детектор витальности лиц работает как на GPU-, так и на CPU-ускорении, однако на CPU он гораздо медленнее.

В этом разделе:

- Включение детектора витальности лиц
- Настройка порога *Liveness*

Включение детектора витальности лиц

Для включения детектора витальности лиц выполните следующие действия:

- Откройте файл конфигурации `/opt/findface-multi/configs/findface-video-worker/findface-video-worker.yaml`. В разделе `liveness` укажите модели нейронных сетей, как показано в примере:

GPU

```
sudo vi /opt/findface-multi/configs/findface-video-worker/findface-video-worker.yaml

liveness:
  fnk: /usr/share/findface-data/models/faceattr/liveness.pacs.v2.gpu.fnk
  norm: /usr/share/findface-data/models/facenorm/facenorm.multicrop_full_crop2x_
  ↪size400.gpu.fnk
  ...
```

CPU

```
sudo vi /opt/findface-multi/configs/findface-video-worker/findface-video-worker.yaml

liveness:
  fnk: /usr/share/findface-data/models/faceattr/liveness.pacs.v2.cpu.fnk
  norm: /usr/share/findface-data/models/facenorm/facenorm.multicrop_full_crop2x_
  ↪size400.cpu.fnk
  ...
```

2. Перезапустите контейнер `findface-multi-findface-video-worker-1`.

```
sudo docker container restart findface-multi-findface-video-worker-1
```

Настройка порога Liveness

При необходимости вы можете настроить пороговое значение для `liveness` в файле конфигурации `/opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py`. Детектор витальности лиц оценивает витальность лица с определенной степенью достоверности. В зависимости порогового значения достоверности, он возвращает бинарный результат **истинное лицо** или **изображение**.

Примечание: Значение по умолчанию является оптимальным. Перед изменением порога проконсультируйтесь у наших специалистов по адресу support@ntechlab.com.

```
sudo vi /opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py

FFSECURITY = {
  ...
  # feature specific confidence thresholds
  'LIVENESS_THRESHOLD': 0.674, # model: [liveness.pacs.v2]
  ...
}
```

Перезапустите контейнер `findface-multi-findface-multi-legacy-1`, если вы меняли порог для `liveness`.

```
sudo docker container restart findface-multi-findface-multi-legacy-1
```

2.5.9 Миграция векторов признаков на другую модель нейронной сети

Совет: Не стесняйтесь обращаться к нашим специалистам по вопросам миграции по адресу support@ntechlab.com.

Важно: В случае если вы выполняете миграцию как часть обновления системы до новой версии, выполните сначала *обновление*. Только после этого приступайте к миграции.

Этот раздел посвящен миграции векторов признаков объектов на другую модель нейронной сети.

Выполните следующие действия:

1. Создайте резервную копию базы данных векторов признаков на основе Tarantool в любой выбранной директории, например, `/etc/ffmulti_dump`.

```
sudo docker exec -it findface-multi-findface-sf-api-1 bash -c "mkdir ffmulti_dump;
↪cd ffmulti_dump && /storage-api-dump -config /etc/findface-sf-api.ini"
sudo docker cp findface-multi-findface-sf-api-1:/ffmulti_dump /etc
```

2. Создайте новые шарды, в которых будут храниться заново сгенерированные векторы признаков.
 1. Перейдите в директорию `/opt/findface-multi/configs/findface-tarantool-server/` и подсчитайте количество шардов по количеству файлов конфигурации `shard-00*.lua`.

Примечание: В примере ниже восемь шардов.

```
cd /opt/findface-multi/configs/findface-tarantool-server

ls -l

shard-001.lua
shard-002.lua
shard-003.lua
shard-004.lua
shard-005.lua
shard-006.lua
shard-007.lua
shard-008.lua
```

2. В директории `/opt/findface-multi/configs/findface-tarantool-server/` создайте новые шарды в том же количестве. Для этого скопируйте уже существующие файлы конфигурации `shard-00*.lua`.

Примечание: Для удобства в качестве второй цифры в новых именах используется 1: `shard-01*.lua`.

```
for i in {1..8}; do sudo cp shard-00$i.lua shard-01$i.lua; done
```

3. В файле конфигурации каждого шарда измените следующие строки в зависимости от его имени:

Прежнее значение	Новое значение
listen = '127.0.0.1:32001'	Listen = '127.0.0.1:32011'
FindFace.start(«127.0.0.1», 8101, {	FindFace.start(«127.0.0.1», 8111, {

Это можно сделать, выполнив следующую команду:

```
for i in {1..8}; do sudo sed -i "s/    listen = '127.0.0.1:3200$i',/    listen
↪= '127.0.0.1:3201$i',/" shard-01$i.lua && sudo sed -i "s/FindFace.start(\"127.
↪0.0.1\", 810$i, {/FindFace.start(\"127.0.0.1\", 811$i, {/" shard-01$i.lua;
↪done
```

4. Создайте директории, в которых будут храниться файлы новых шардов.

```
cd /opt/findface-multi/data/findface-tarantool-server

sudo mkdir -p shard-01{1..8}/{index,snapshots,xlogs}
```

3. Откройте файл конфигурации /opt/findface-multi/configs/findface-extraction-api/findface-extraction-api.yaml и замените модели extraction на новые в параметрах body_emben, car_emben и face_emben, в зависимости от мигрируемых типов объектов.

```
sudo vi /opt/findface-multi/configs/findface-extraction-api/findface-extraction-api.
↪yaml

extractors:
  ...
  models:
    ...
    body_emben: pedrec/<new_model_body>.cpu<gpu>.fnk
    ...
    car_emben: carrec/<new_model_car>.cpu<gpu>.fnk
    ...
    face_emben: face/<new_model_face>.cpu<gpu>.fnk
    ...
```

Перезапустите контейнер findface-multi-findface-extraction-api-1.

```
cd /opt/findface-multi/
sudo docker-compose restart findface-extraction-api
```

4. В файле docker-compose.yaml создайте новые сервисы для каждого нового шарда. Для этого скопируйте существующий сервис и замените имя шарда на новое.

```
sudo vi docker-compose.yaml

services:
  ...
  findface-tarantool-server-shard-**011**:
    depends_on: [findface-ntls]
    image: docker.int.ntl/ntech/universe/tntapi:ffserver-8.221216
    network_mode: service:pause
    restart: always
    volumes: [ './configs/findface-tarantool-server/shard-**011*.lua:/etc/tarantool/
```

(continues on next page)

(продолжение с предыдущей страницы)

```

↪instances.enabled/FindFace.lua:ro',
    './data/findface-tarantool-server/shard-**011**:/var/lib/tarantool/FindFace',
↪'./configs/findface-tarantool-server/tnt-schema.lua:/tnt_schema.lua:ro']
...

```

5. Запустите новые шарды, подняв контейнеры.

```
sudo docker-compose up -d
```

6. На основании приведенного ниже примера создайте файл конфигурации с настройками миграции migration.yaml.

```

sudo vi migration.yaml

extraction-api:
  timeouts:
    connect: 5s
    response_header: 30s
    overall: 35s
    idle_connection: 0s
  extraction-api: http://127.0.0.1:18666
storage-api-from: # current location of the gallery
  timeouts:
    connect: 5s
    response_header: 30s
    overall: 35s
    idle_connection: 10s
  max-idle-conns-per-host: 20
  shards:
    - master: http://127.0.0.1:8101/v2/
      slave: ''
    - master: http://127.0.0.1:8102/v2/
      slave: ''
    - master: http://127.0.0.1:8103/v2/
      slave: ''
    - master: http://127.0.0.1:8104/v2/
      slave: ''
    - master: http://127.0.0.1:8105/v2/
      slave: ''
    - master: http://127.0.0.1:8106/v2/
      slave: ''
    - master: http://127.0.0.1:8107/v2/
      slave: ''
    - master: http://127.0.0.1:8108/v2/
      slave: ''
  storage-api-to:
    timeouts:
      connect: 5s
      response_header: 30s
      overall: 35s
      idle_connection: 10s
    max-idle-conns-per-host: 20

```

(continues on next page)

(продолжение с предыдущей страницы)

```

shards:
  - master: http://127.0.0.1:8111/v2/
    slave: ''
  - master: http://127.0.0.1:8112/v2/
    slave: ''
  - master: http://127.0.0.1:8113/v2/
    slave: ''
  - master: http://127.0.0.1:8114/v2/
    slave: ''
  - master: http://127.0.0.1:8115/v2/
    slave: ''
  - master: http://127.0.0.1:8116/v2/
    slave: ''
  - master: http://127.0.0.1:8117/v2/
    slave: ''
  - master: http://127.0.0.1:8118/v2/
    slave: ''
workers_num: 3
faces_limit: 100
extraction_batch_size: 8
normalized_storage:
  type: webdav
  enabled: True
  webdav:
    upload-url: http://127.0.0.1:3333/uploads/
  s3:
    endpoint: ''
    bucket-name: ''
    access-key: ''
    secret-access-key: ''
    secure: False
    region: ''
    public-url: ''
    operation-timeout: 30

```

В разделе `storage-api-from` укажите шарды, с которых осуществляется миграция.

```

storage-api-from: # current location of the gallery
...
shards:
  - master: http://127.0.0.1:8101/v2/
    slave: ''
  - master: http://127.0.0.1:8102/v2/
    slave: ''
  - master: http://127.0.0.1:8103/v2/
    slave: ''
  - master: http://127.0.0.1:8104/v2/
    slave: ''
  - master: http://127.0.0.1:8105/v2/
    slave: ''
  - master: http://127.0.0.1:8106/v2/
    slave: ''

```

(continues on next page)

(продолжение с предыдущей страницы)

```
- master: http://127.0.0.1:8107/v2/
  slave: ''
- master: http://127.0.0.1:8108/v2/
  slave: ''
...
```

В разделе `storage-api-to` укажите новые шарды, в которых будут храниться данные после миграции.

```
storage-api-to:
...
shards:
- master: http://127.0.0.1:8111/v2/
  slave: ''
- master: http://127.0.0.1:8112/v2/
  slave: ''
- master: http://127.0.0.1:8113/v2/
  slave: ''
- master: http://127.0.0.1:8114/v2/
  slave: ''
- master: http://127.0.0.1:8115/v2/
  slave: ''
- master: http://127.0.0.1:8116/v2/
  slave: ''
- master: http://127.0.0.1:8117/v2/
  slave: ''
- master: http://127.0.0.1:8118/v2/
  slave: ''
...
```

7. Скопируйте файл `migration.yaml` внутрь контейнера `findface-multi-findface-sf-api-1`. Запустите утилиту `sf-api-migrate` с опцией `-config` и укажите файл конфигурации `migration.yaml`.

```
sudo docker cp migration.yaml findface-multi-findface-sf-api-1:/
sudo docker exec findface-multi-findface-sf-api-1 ./sf-api-migrate -config migration.yaml
```

Примечание: Если в системе много событий и карточек, процесс миграции может занять значительное количество времени.

8. После завершения миграции удалите сервисы старых шардов из файла `docker-compose.yaml` и остановите используемые ими контейнеры.

```
sudo docker-compose up -d --remove-orphans
```

9. Откройте файл конфигурации `/opt/findface-multi/configs/findface-sf-api/findface-sf-api.yaml` и измените порты шардов в зависимости от новых настроек. Перезапустите контейнер `findface-multi-findface-sf-api-1`.

```
sudo vi /opt/findface-multi/configs/findface-sf-api/findface-sf-api.yaml
```

(continues on next page)

(продолжение с предыдущей страницы)

```
storage-api:
  shards:
    - master: http://127.0.0.1:8111/v2/
      slave: ''
    - master: http://127.0.0.1:8112/v2/
      slave: ''
    - master: http://127.0.0.1:8113/v2/
      slave: ''
    - master: http://127.0.0.1:8114/v2/
      slave: ''
    - master: http://127.0.0.1:8115/v2/
      slave: ''
    - master: http://127.0.0.1:8116/v2/
      slave: ''
    - master: http://127.0.0.1:8117/v2/
      slave: ''
    - master: http://127.0.0.1:8118/v2/
      slave: ''

sudo docker-compose restart findface-sf-api
```

10. Мигрируйте кластеры, если *данная функция* включена в системе. Для этого выполните следующую команду:

Примечание: Перечислите мигрируемые типы объектов в виде опций команды: `--face`, `--body`, `--car`.

```
sudo docker exec -it findface-multi-findface-multi-legacy-1 /opt/findface-security/
↪bin/python3 /tigre_prototype/manage.py migrate_clusters --face --body --car --use-
↪best-event --use-thumbnail --force-clustering
```

В результате система повторно сгенерирует векторы признаков для существующих кластерных событий и автоматически запустит запланированную кластеризацию для пересборки кластеров.

2.5.10 Деактивация сервиса `findface-liveness-api`, установленного вместе с FindFace Multi

Если вы не используете сервис `findface-liveness-api`, установленный вместе с FindFace Multi, и аутентификация по лицу не актуальна для вашей системы, рекомендуется деактивировать `findface-liveness-api`.

Выполните следующие действия:

1. Остановите контейнер `findface-multi-findface-liveness-api-1` и отключите его автозагрузку, выполнив следующую последовательность команд:

```
sudo docker container stop findface-multi-findface-liveness-api-1
sudo docker update --restart=no findface-multi-findface-liveness-api-1
```

2. Откройте файл конфигурации `/opt/findface-multi/configs/findface-extraction-api/findface-extraction-api.yaml`.


```
sudo vi /opt/findface-multi/configs/findface-extraction-api/findface-extraction-api.  
↪yaml
```

- Отключите модель нейронной сети, используемую сервисом `findface-liveness-api`, передав пустое значение `''` в параметры `multicrop_full_crop2x` и `face_liveness`.

Примечание: Не удаляйте сам параметр, поскольку в этом случае будет выполняться поиск модели по умолчанию.

```
normalizers:  
  models:  
    ...  
    multicrop_full_crop2x: ''  
    ...  
  
extractors:  
  models:  
    ...  
    face_liveness: ''  
    ...
```

- Перезапустите контейнер `findface-multi-findface-extraction-api-1`.

```
sudo docker container restart findface-multi-findface-extraction-api-1
```

2.6 Обслуживание и устранение неисправностей

2.6.1 Обновление до FindFace Multi 2.0

Совет: Если вы используете наш продукт FindFace Security, развернутый в Ubuntu 18.04, сначала **обновите** его до FindFace Multi 1.2, а затем до FindFace Multi 2.0.

В FindFace Multi 2.0 не реализована работа плагинов Происшествия и VNS. Для этого есть отдельный продукт - [FindFace CIBR](#). Поэтому включать плагины `ffsecurity_incidents`, `ffsecurity_vns` и `ffsecurity_puppeteer` в файле конфигурации `findface-multi-legacy.py` после обновления не нужно.

Интеграция с Axon Next в FindFace Multi 2.0 является частью интеграции с *внешними VMS* и настраивается через плагин интеграции с VMS.

Если в версии FindFace Multi 1.2 вы создавали карточки человека или автомобиля с кастомными метаполями, то, чтобы они продолжали отображаться в интерфейсе FindFace Multi 2.0, необходимо в шаге №2 скопировать секцию `CUSTOM_FIELDS` из старого конфигурационного файла и в шаге №8 перенести ее в новый конфигурационный файл.

Для обновления FindFace Multi с версии 1.2 до версии 2.0 выполните следующие действия:

- Создайте резервную копию старой схемы базы данных Tarantool:

```
sudo cp /etc/findface-security/tnt_schema.lua /etc/findface-security/old_tnt_schema.  
↪lua
```

Одним из самых существенных отличий FindFace Multi 2.0 от предыдущих версий продукта является структура биометрической базы данных Tarantool (так называемая «метасхема»). В новой структуре выделено несколько пространств (spaces), тогда как в прежних версиях было только одно пространство (space) по умолчанию.

2. Откройте файл конфигурации `/etc/findface-security/config.py`. Сохраните значения параметров `EXTERNAL_ADDRESS`, `SECRET_KEY`, `VIDEO_DETECTOR_TOKEN`, `ROUTER_URL`, `CUSTOM_FIELDS` для последующего использования.

```
sudo vi /etc/findface-security/config.py

EXTERNAL_ADDRESS = "http://172.20.77.58"

...
# use pwgen -sncy 50 1/tr "" "." to generate your own unique key
SECRET_KEY = 'c8b533847bbf7142102de1349d33a1f6'
FFSECURITY = {
  'VIDEO_DETECTOR_TOKEN': '381b0f4a20495227d04185ab02f5085f',
  ...
  'ROUTER_URL': 'http://172.20.77.58',
  ...
  # -- Custom model fields --
  # Edit CUSTOM_FIELDS -> `human_card` section to customize human card fields.
  # Edit CUSTOM_FIELDS -> `face_object` section to customize face object fields.
  # Below is an example with every field type possible.
  # 'CUSTOM_FIELDS': {
  #   'human_card': {
  #     'items': [
  #       {
  #         'name': 'personid',
  #         'default': '',
  #         'label': 'PersonID',
  #         'display': ['list', 'form'],
  #         'description': 'Sigur person ID',
  #         'editable': False
  #       },
  #       {
  #         'name': 'firstname',
  #         'default': '',
  #         'label': 'First Name',
  #         'display': ['list', 'form'],
  #         'description': 'Sigur first name',
  #         'editable': False
  #       },
  #       {
  #         'name': 'lastname',
  #         'default': '',
  #         'label': 'Last Name',
  #         'display': ['list', 'form'],
  #         'description': 'Sigur last name',
  #         'editable': False
  #       },
  #       {
  #         'name': 'version',
  #         'default': '',
```

(continues on next page)

(продолжение с предыдущей страницы)

```

#           'label': 'Version',
#           'display': ['list', 'form'],
#           'description': 'Sigur photo version',
#           'editable': False
#       }
#   ],
#   'filters': [
#       {
#           'name': 'personid',
#           'label': 'Sigur person ID filter',
#           'field': 'personid'
#       }
#   ]
# },
# 'face_object': {
#   'items': [
#       {
#           "field_name": "tag_name_1",
#           "type": "string",
#           "default": "change_me"
#       },
#       {
#           "field_name": "tag_name_2",
#           "type": "uint",
#           "default": 123
#       },
#       {
#           "field_name": "tag_name_3",
#           "type": "bool",
#           "default": True
#       }
#   ]
# }
# },
# }

```

3. Остановите сервис `findface-security`.

```
sudo systemctl stop findface-security.service
```

4. Создайте резервную копию базы данных векторов признаков на основе Tarantool в любой выбранной директории, например, `/etc/findface_dump`.

```

sudo mkdir -p /etc/findface_dump
cd /etc/findface_dump
sudo findface-storage-api-dump -config /etc/findface-sf-api.ini

```

5. Перед установкой новой версии остановите и отключите все сервисы во избежание конфликта портов.

Примечание: В примере ниже восемь шардов. Если количество шардов в вашей системе отличается от количества шардов в примере, измените приведенную ниже команду соответствующим образом. Например, для системы с шестнадцатью шардами замените `tarantool@shard-00{1..8}`.

service на tarantool@shard-0{01..16}.service. Список активных шардов можно посмотреть с помощью команды `ls /etc/tarantool/instances.enabled/`.

Можно остановить и отключить сервисы один за другим:

```
sudo systemctl stop postgresql.service
sudo systemctl stop postgresql@10-main
sudo systemctl stop findface-*.service
sudo systemctl stop pgbouncer.service
sudo systemctl stop tarantool@shard-00{1..8}.service
sudo systemctl stop nats-server.service
sudo systemctl stop etcd.service
sudo systemctl stop mongod.service
sudo systemctl stop mongod.service
sudo systemctl stop memcached.service
sudo systemctl stop nginx.service

sudo systemctl disable postgresql.service
sudo systemctl disable postgresql@10-main
sudo systemctl disable pgbouncer.service
sudo systemctl disable findface-extraction-api.service
sudo systemctl disable findface-security.service
sudo systemctl disable findface-security-onvif.service
sudo systemctl disable findface-sf-api.service
sudo systemctl disable findface-ntls.service
sudo systemctl disable findface-video-manager.service
sudo systemctl disable findface-video-worker-cpu.service
sudo systemctl disable findface-video-worker-gpu.service
sudo systemctl disable findface-counter.service
sudo systemctl disable findface-liveness-api.service
sudo systemctl disable findface-video-streamer-cpu.service
sudo systemctl disable findface-video-streamer-gpu.service
sudo systemctl disable findface-video-storage.service
sudo systemctl disable tarantool@shard-00{1..8}.service
sudo systemctl disable nats-server.service
sudo systemctl disable etcd.service
sudo systemctl disable mongod.service
sudo systemctl disable mongod.service
sudo systemctl disable memcached.service
sudo systemctl disable nginx.service
```

Или использовать более компактные команды:

```
sudo systemctl stop postgresql.service postgresql@10-main findface-*.service
↪pgbouncer.service tarantool@shard-00{1..8}.service nats-server.service etcd.
↪service mongod.service mongod.service memcached.service nginx.service

sudo systemctl disable postgresql.service postgresql@10-main pgbouncer.service
↪findface-extraction-api.service findface-security.service findface-security-onvif.
↪service findface-sf-api.service findface-ntls.service findface-video-manager.
↪service findface-video-worker-cpu.service findface-video-worker-gpu.service
↪findface-counter.service findface-liveness-api.service findface-video-streamer-
↪cpu.service findface-video-streamer-gpu.service findface-video-storage.service
```

(continues on next page)

(продолжение с предыдущей страницы)

```
↪tarantool@shard-00{1..8}.service nats-server.service etcd.service mongod.service
↪mongodb.service memcached.service nginx.service
```

6. *Установите* экземпляр FindFace Multi 2.0. Не забудьте предварительно подготовить сервер:

См.:

- *Подготовка сервера Ubuntu*
- *Подготовка сервера CentOS 7*
- *Подготовка сервера Debian 11*

7. После установки продукта остановите все контейнеры FindFace Multi из директории `/opt/findface-multi/`.

```
cd /opt/findface-multi/
sudo docker-compose stop
```

8. Откройте файл конфигурации `/opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py` и вставьте сохраненные на шаге №2 значения для параметров `EXTERNAL_ADDRESS`, `SECRET_KEY`, `VIDEO_DETECTOR_TOKEN`, `ROUTER_URL`, и `CUSTOM_FIELDS`.

```
sudo vi /opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py
...
# Use pwgen -sncy 50 1/tr "" "." to generate your own unique key
SECRET_KEY = '002231ccb690586f4d33e98322c591bb'
...
SERVICE_EXTERNAL_ADDRESS = 'http://172.20.77.58'
# EXTERNAL_ADDRESS is used to access objects created inside FFSecurity via external
↪links.
EXTERNAL_ADDRESS = 'http://172.20.77.58'
...
# findface-video-worker authorization token
'VIDEO_DETECTOR_TOKEN': '8977e1b0067d43f6c908d0bf60363255',
...
# findface-video-worker face posting address,
# it must be set to either FFSecurity EXTERNAL_ADDRESS (by default)
# or findface-facerouter url (in some specific cases)
'ROUTER_URL': 'http://127.0.0.1:80',
```

9. Откройте старую версию файла конфигурации `findface-ntls`, доступную по адресу `/etc/findface-ntls.cfg`, и сравните ее с новой версией `/opt/findface-multi/configs/findface-ntls/findface-ntls.yaml`. Перенесите все пользовательские параметры из старой версии в новую. Сделайте то же самое для других компонентов, например, для `findface-extract-api` проверьте `/etc/findface-extract-api.ini` на соответствие `/opt/findface-multi/configs/findface-extraction-api/findface-extraction-api.yaml`, для `findface-sf-api` проверьте `/etc/findface-sf-api.ini` на соответствие `/opt/findface-multi/configs/findface-sf-api/findface-sf-api.yaml` и т. д.

```
sudo vi /etc/findface-ntls.cfg
sudo vi /opt/findface-multi/configs/findface-ntls/findface-ntls.yaml
sudo vi /etc/findface-extraction-api.ini
sudo vi /opt/findface-multi/configs/findface-extraction-api/findface-extraction-api.
↪yaml
```

(continues on next page)

(продолжение с предыдущей страницы)

```
sudo vi /etc/findface-sf-api.ini
sudo vi /opt/findface-multi/configs/findface-sf-api/findface-sf-api.yaml
```

Важно: Правила переноса параметров из старого файла конфигурации в новый:

- Если в новом конфигурационном файле на месте старой модели нейронной сети указана новая модель, замените новую модель на ту, которая использовалась ранее (при условии, что предыдущая модель все еще включена в FindFace Multi 2.0), а если предыдущая модель отсутствует в FindFace Multi 2.0, то ничего не меняйте. В этом случае вам придется выполнить *миграцию на другую модель нейронной сети*.
- Если параметр в старом файле конфигурации имел пустое значение, а в новом файле конфигурации он заполнен – сотрите это значение.
- Оставьте как есть параметры, не включенные в старый файл конфигурации, но присутствующие в новом файле конфигурации.

10. Измените структуру базы данных Tarantool, применив схему `tnt_schema.lua` из версии Findface Multi 2.0.

```
sudo docker run --rm --network host --volume '/opt/findface-multi/configs/findface-
↳multi-legacy/findface-multi-legacy.py:/etc/findface-security/config.py:ro' docker.
↳int.ntl/ntech/multi/multi/legacy:ffmulti-2.0.0 make-tnt-schema | sudo tee /etc/
↳findface-security/tnt_schema.lua
```

11. Удалите данные из каталогов, соответствующих активным шардам.

```
sudo rm /opt/ntech/var/lib/tarantool/shard-*/{index,snapshots,xlogs}/*
```

12. Скопируйте метасхему пространства (space) `default` из старого файла конфигурации `old_tnt_schema.lua` в новый файл `tnt_schema.lua`, чтобы старая метасхема продолжала быть доступной. Это можно сделать следующим простым способом:

- 12.1. В файле `/etc/findface-security/old_tnt_schema.lua` переименуйте следующие поля:

```
meta_scheme --> meta_scheme_default
meta_indexes --> meta_indexes_default
```

- 12.2. В новом файле конфигурации `/etc/findface-security/tnt_schema.lua` замените строки в начале файла:

```
spaces = {
  default = {
    meta_scheme = {
      -- internal.normalized_id:
      {
        default = '',
        field_type = 'string',
        id = 1,
        name = 'normalized_id',
      },
      -- internal.feat:
      {
```

(continues on next page)

(продолжение с предыдущей страницы)

```

        default = '',
        field_type = 'string',
        id = 2,
        name = 'feat',
    },
},
meta_indexes = {}
},

```

на следующие:

```

dofile("/etc/findface-security/old_tnt_schema.lua")
spaces = {
    default = {
        meta_scheme=meta_scheme_default,
        meta_indexes=meta_indexes_default
    },

```

13. Перейдите в директорию с файлами конфигурации Tarantool `/etc/tarantool/instances.available/`. Импортируйте новую метасхему `tnt_schema.lua` в каждый файл конфигурации `shard-00*.lua`, как показано в примере.

```

sudo vi /etc/tarantool/instances.available/shard-00*.lua

dofile("/etc/findface-security/tnt_schema.lua")
FindFace = require("FindFace")
FindFace.start("127.0.0.1", 8104, {
    license_ntls_server="127.0.0.1:3133",
    replication = replication_master,
    spaces = spaces
})

```

14. Удалите генерируемый пакетом `findface-tarantool-server` дефолтный конфигурационный файл `FindFace.lua` (он мешает сделать перезагрузку).

```

sudo rm -rf /etc/tarantool/instances*/FindFace.lua

```

15. Перезапустите шарды `findface-tarantool-server`.

```

TNT=$(ls /etc/tarantool/instances.enabled/ | cut -c 7,8,9)
for i in $TNT; do sudo systemctl restart tarantool@shard-$i.service ; done

```

После выполненных операций в шардах по-прежнему останутся старые галереи, созданные в пространстве `default`. При этом станут доступны новые пространства (например, `ffsec_body_objects_space`, `ffsec_face_clusters_space` и т. д.).

16. Восстановите в базе старые данные из резервной копии. Данные будут восстановлены в том виде, в котором они существовали в предыдущей версии: все галереи останутся в пространстве `default`.

```

sudo systemctl start findface-ntls.service
cd /etc/findface_dump
for x in *.json; do sudo findface-storage-api-restore -config /etc/findface-sf-api.
→ini < "$x"; done

```

17. Выполните миграцию галерей из пространства `default` на новые пространства:

```
sudo systemctl start findface-sf-api.service
sudo systemctl start nginx.service
sudo docker run --rm --network host --volume '/opt/findface-multi/configs/findface-
↳multi-legacy/findface-multi-legacy.py:/etc/findface-security/config.py:ro' docker.
↳int.ntl/ntech/multi/multi/legacy:ffmulti-2.0.0 /opt/findface-security/bin/python3
↳/tigre_prototype/manage.py migrate_tnt_space
```

18. Выполните миграции базы данных PostgreSQL для совместимости с FindFace Multi 2.0. Сделайте следующее:

18.1. Перейдите в файл конфигурации `/opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py`. В секции `DATABASES` -> `default` временно поменяйте пароль `PASSWORD` на старый, используемый в конфигурационном файле `/etc/findface-security/config.py`.

Важно: Обязательно сохраните пароль из файла конфигурации `/opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py`. Позже он понадобится.

```
sudo vi /opt/findface-multi/configs/findface-multi-legacy/findface-multi-
↳legacy.py

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'DISABLE_SERVER_SIDE_CURSORS': True,
        'NAME': 'ffsecurity', 'HOST': '127.0.0.1', 'PORT': 5439, 'USER':
↳'ntech', 'PASSWORD': 'XXXXXXXXXXXXXXXXX'
    }
}
```

- 18.2. В файле `/etc/pgbouncer/pgbouncer.ini` добавьте следующую строку в секцию `databases`:

```
ffsecurity_session = dbname=ffsecurity host=localhost port=5432 user=ntech
↳pool_mode=session pool_size=10
```

- 18.3. Выполните миграции базы данных на хосте:

```
sudo systemctl start postgresql.service
sudo systemctl start pgbouncer.service
sudo docker run --rm --network host --volume '/opt/findface-multi/configs/
↳findface-multi-legacy/findface-multi-legacy.py:/etc/findface-security/
↳config.py:ro' docker.int.ntl/ntech/multi/multi/legacy:ffmulti-2.0.0 /opt/
↳findface-security/bin/python3 /tigre_prototype/manage.py migrate
```

- 18.4. Сделайте резервное копирование существующей базы данных с PostgreSQL, установленной на хост-системе.

```
cd /opt/findface-multi/
sudo -u postgres pg_dump --verbose --disable-triggers ffsecurity | sudo tee
↳dump_ffsecurity.sql
```

(continues on next page)

(продолжение с предыдущей страницы)

```

sudo -u postgres pg_dump -t auth_group -t ffsecurity_adgroupguid -t
↪ffsecurity_deviceblacklistrecord -t ffsecurity_ffsecauthsession -t
↪ffsecurity_grouppermission -t ffsecurity_runtimesetting -t ffsecurity_
↪user -t ffsecurity_user_groups -t ffsecurity_user_user_permissions -t
↪ffsecurity_userkeyvalue -t Knox_authtoken -t ffsecurity_
↪watchlistpermission -t ffsecurity_cameragrouppermission --data-only --
↪verbose --no-acl --no-owner --disable-triggers ffsecurity | sudo tee dump_
↪identity_provider.sql

```

18.5. Сделайте резервное копирование прав доступа ролей.

```

sudo docker run --rm --network host --volume '/opt/findface-multi/configs/
↪findface-multi-legacy/findface-multi-legacy.py:/etc/findface-security/
↪config.py:ro' docker.int.ntl/ntech/multi/multi/legacy:ffmulti-2.0.0 /opt/
↪findface-security/bin/python3 /tigre_prototype/manage.py dump_permissions
↪| sudo tee permissions.csv

```

18.6. Поменяйте обратно пароль, измененный на шаге №18.1

18.7. Остановите все сервисы.

```

sudo systemctl stop findface-sf-api.service nginx.service tarantool@shard-00
↪{1..8}.service postgresql.service pgbouncer.service

```

18.8. Откройте файл /opt/findface-multi/docker-compose.yaml и сохраните значение POSTGRES_PASSWORD для использования в последующих командах.

18.9. Пересоздайте базу данных ffsecurity, чтобы очистить её от дефолтных данных. Вставьте значение {POSTGRES_PASSWORD}, которое вы ранее скопировали в шаге №18.8, в команду ниже:

```

sudo docker-compose up -d postgresql
sudo docker exec -it -u postgres findface-multi-postgresql-1 /bin/bash -c
↪"PGPASSWORD={POSTGRES_PASSWORD} dropdb ffsecurity"
sudo docker exec -it -u postgres findface-multi-postgresql-1 /bin/bash -c
↪"PGPASSWORD={POSTGRES_PASSWORD} createdb -O ntech --encoding=UTF-8 --lc-
↪collate=C.UTF-8 --lc-ctype=C.UTF-8 --template=template0 ffsecurity"

```

18.10. Восстановите данные в только что созданную базу ffsecurity. Вставьте значение {POSTGRES_PASSWORD}, которое вы ранее скопировали в шаге №18.8, в команду ниже:

```

sudo docker exec -i findface-multi-postgresql-1 /bin/bash -c "PGPASSWORD=
↪{POSTGRES_PASSWORD} psql --username postgres ffsecurity" < dump_
↪ffsecurity.sql

```

18.11. Пересоздайте базу данных ffsecurity_identity_provider, чтобы очистить её от дефолтных данных. Вставьте значение {POSTGRES_PASSWORD}, которое вы ранее скопировали в шаге №18.8, в команду ниже:

```

sudo docker exec -it -u postgres findface-multi-postgresql-1 /bin/bash -c
↪"PGPASSWORD={POSTGRES_PASSWORD} dropdb ffsecurity_identity_provider"
sudo docker exec -it -u postgres findface-multi-postgresql-1 /bin/bash -c
↪"PGPASSWORD={POSTGRES_PASSWORD} createdb -O ntech --encoding=UTF-8 --lc-
↪collate=C.UTF-8 --lc-ctype=C.UTF-8 --template=template0 ffsecurity_
↪identity_provider"

```

18.12. Запустите миграции.

```
sudo docker-compose up -d pgbouncer
sudo docker run --rm --network host --volume '/opt/findface-multi/configs/
↳ findface-multi-identity-provider/findface-multi-identity-provider.py:/etc/
↳ findface-security/config.py:ro' docker.int.ntl/ntech/multi/multi/identity-
↳ provider:ffmulti-2.0.0 /opt/findface-security/bin/python3 /tigre_
↳ prototype/manage.py migrate
```

18.13. Восстановите данные в базу ffsecurity_identity_provider. Вставьте значение {POSTGRES_PASSWORD}, которое вы ранее скопировали в шаге №18.8, в команду ниже:

```
sudo docker exec -i findface-multi-postgresql-1 /bin/bash -c "PGPASSWORD=
↳ {POSTGRES_PASSWORD} psql --username postgres ffsecurity_identity_provider
↳ < dump_identity_provider.sql
```

18.14. Запустите все сервисы.

```
sudo docker-compose up -d
```

18.15. Запустите команду создания записей в таблице outbox для списков наблюдения и групп камер. Вставьте значение {POSTGRES_PASSWORD}, которое вы ранее скопировали в шаге №18.8, в команду ниже:

```
sudo docker exec -i findface-multi-postgresql-1 /bin/bash -c "PGPASSWORD=
↳ {POSTGRES_PASSWORD} pg_dump --username postgres -f cg_wl_permissions.sql -
↳ t ffsecurity_cameragrouppermission -t ffsecurity_watchlistpermission --
↳ data-only ffsecurity_identity_provider"
sudo docker run --rm --network host --volume '/opt/findface-multi/configs/
↳ findface-multi-legacy/findface-multi-legacy.py:/etc/findface-security/
↳ config.py:ro' docker.int.ntl/ntech/multi/multi/legacy:ffmulti-2.0.0 /opt/
↳ findface-security/bin/python3 /tigre_prototype/manage.py create_outbox
sudo docker exec -i findface-multi-postgresql-1 /bin/bash -c "PGPASSWORD=
↳ {POSTGRES_PASSWORD} psql --username postgres -c 'TRUNCATE ffsecurity_
↳ cameragrouppermission, ffsecurity_watchlistpermission RESTART IDENTITY;'
↳ ffsecurity_identity_provider"
sudo docker exec -i findface-multi-postgresql-1 /bin/bash -c "PGPASSWORD=
↳ {POSTGRES_PASSWORD} psql --username postgres ffsecurity_identity_provider
↳ < cg_wl_permissions.sql"
```

18.16. Прежде чем восстановить права доступа ролей в сервис identity_provider, проверьте файл /opt/findface-multi/permissions.csv. Обязательно замените *_ffsecauthtoken на *_authtoken, если таковой имеется. Чаще всего это применимо к тем случаям, когда установка FindFace Multi 1.2 представляла собой обновление с более ранних версий продукта.

После этого восстановите права доступа ролей в сервис identity_provider.

```
sudo docker run --rm --network host -v /opt/findface-multi/configs/findface-
↳ multi-identity-provider/findface-multi-identity-provider.py:/etc/findface-
↳ security/config.py:ro -v $(pwd)/permissions.csv:/var/permissions.csv:ro
↳ docker.int.ntl/ntech/multi/multi/identity-provider:ffmulti-2.0.0 /opt/
↳ findface-security/bin/python3 /tigre_prototype/manage.py load_permissions
↳ /var/permissions.csv
```

18.17. Скопируйте полные кадры, нормализованные изображения и лицензионный файл. Скопируйте файлы из папки /opt/ntech/license/ в папку /opt/findface-multi/data/

findface-ntls/, из /var/lib/findface-security/uploads/ в /opt/findface-multi/data/findface-multi-legacy/uploads/, из /var/lib/ffupload/uploads/ в /opt/findface-multi/data/findface-upload/uploads/.

```
sudo cp -r /opt/ntech/license/* /opt/findface-multi/data/findface-ntls/
sudo cp -r /var/lib/findface-security/uploads/* /opt/findface-multi/data/
↪findface-multi-legacy/uploads/
sudo cp -r /var/lib/ffupload/uploads/* /opt/findface-multi/data/findface-
↪upload/uploads/
sudo chmod 777 -R /opt/findface-multi/data/findface-upload/uploads/
sudo chown www-data:www-data -R /opt/findface-multi/data/findface-upload/
↪uploads/*
```

Также можно напрямую смонтировать указанные выше папки в соответствующие контейнеры через файл docker-compose.yaml, как показано в примере ниже:

```
sudo vi /opt/findface-multi/docker-compose.yaml

findface-upload:
  image: docker.int.ntl/ntech/universe/upload:ffserver-8.221216
  logging: {driver: journald}
  network_mode: service:pause
  restart: always
  volumes: ['./configs/findface-upload/40-ffupload.sh:/docker-entrypoint.d/
↪40-ffupload.sh:ro',
            '/var/lib/ffupload:/var/lib/ffupload']
findface-multi-identity-provider:
  depends_on: [pgbouncer, nats, findface-sf-api, findface-liveness-api,
↪etcd]
  environment: {ADMIN_PASSWORD: <ADMIN_PASSWORD>}
  image: docker.int.ntl/ntech/multi/multi/identity-provider:ffmulti-2.0.0
  logging: {driver: journald}
  network_mode: service:pause
  restart: always
  volumes: ['./configs/findface-multi-identity-provider/findface-multi-
↪identity-provider.py:/etc/findface-security/config.py:ro',
            '/var/lib/findface-security/uploads:/var/lib/findface-security/uploads']
findface-multi-legacy:
  depends_on: [pgbouncer, nats, findface-sf-api, findface-counter, findface-
↪liveness-api, etcd]
  environment: {ADMIN_PASSWORD: <ADMIN_PASSWORD>}
  image: docker.int.ntl/ntech/multi/multi/legacy:ffmulti-2.0.0
  logging: {driver: journald}
  network_mode: service:pause
  restart: always
  volumes: ['./configs/findface-multi-legacy/findface-multi-legacy.py:/etc/
↪findface-security/config.py:ro',
            '/var/lib/findface-security/uploads:/var/lib/findface-security/uploads']
findface-multi-ui:
  depends_on: [findface-multi-legacy]
  image: docker.int.ntl/ntech/multi/multi/ui:ffmulti-2.0.0
  logging: {driver: journald}
  network_mode: service:pause
  restart: always
```

(continues on next page)

(продолжение с предыдущей страницы)

```
volumes: ['./configs/findface-multi-ui/nginx-site.conf:/etc/nginx/conf.d/
↪default.conf:ro',
  '/var/lib/findface-security/uploads:/var/lib/findface-security/uploads']
```

18.18. Чтобы перенести данные Tarantool, выполните следующие действия:

Остановите все контейнеры FindFace Multi:

```
sudo docker-compose down
```

Запустите снова старые шарды и сервис findface-sf-api :

```
sudo systemctl start tarantool@shard-00{1..8}.service findface-sf-api.
↪service
```

Создайте новую резервную копию базы данных векторов признаков:

```
sudo mkdir -p /etc/findface_dump_final
sudo findface-storage-api-dump -output-dir=/etc/findface_dump_final -config
↪/etc/findface-sf-api.ini
```

Остановите остальные сервисы, очистите директорию instances.enabled, снова запустите контейнеры и выполните storage-api-restore:

```
sudo systemctl stop tarantool@shard-00{1..8}.service findface-sf-api.
↪service findface-ntls.service
sudo rm /etc/tarantool/instances.enabled/*
sudo docker-compose up -d
sudo findface-storage-api-restore -config /opt/findface-multi/configs/
↪findface-sf-api/findface-sf-api.yaml /etc/findface_dump_final/*.json
```

Обновление завершено, но в новой версии присутствуют новые модели нейронных сетей, поэтому необходимо также сделать *миграцию векторов признаков на другую модель нейронной сети*, либо можно использовать старые нейронные сети, перенеся их из директории /usr/share/findface-data/models/ в директорию /opt/findface-multi/models/, указав их в файле /opt/findface-multi/configs/findface-extraction-api/findface-extraction-api.yaml.

Обратите внимание, что модели обнаружения объектов в FindFace Multi 1.2 хранятся в отдельных директориях, то есть модели обнаружения лиц хранятся в директории /usr/share/findface-data/models/facedet/, модели обнаружения автомобилей и силуэтов хранятся в директориях /usr/share/findface-data/models/cadet/ и /usr/share/findface-data/models/pedet/ соответственно. В то время как в FindFace Multi 2.0 все модели обнаружения объектов можно найти в одной директории /opt/findface-multi/models/detector/. При переносе старых моделей нейронных сетей из директории /usr/share/findface-data/models/ в директорию /opt/findface-multi/models/ поместите все модели обнаружения объектов (facedet, cadet, pedet) в директорию /opt/findface-multi/models/detector/.

См. для справки:

```
$ ls -lash /usr/share/findface-data/models
total 52K
4.0K drwxr-xr-x 13 root root 4.0K Jul 15 14:48 .
4.0K drwxr-xr-x  3 root root 4.0K Jul 15 14:48 ..
4.0K drwxr-xr-x  2 root root 4.0K Jul 15 14:48 cadet
```

(continues on next page)

(продолжение с предыдущей страницы)

```
4.0K drwxr-xr-x 2 root root 4.0K Jul 15 14:48 carattr
4.0K drwxr-xr-x 2 root root 4.0K Jul 15 14:48 carnorm
4.0K drwxr-xr-x 2 root root 4.0K Jul 15 14:48 carrec
4.0K drwxr-xr-x 2 root root 4.0K Jul 15 14:48 face
4.0K drwxr-xr-x 2 root root 4.0K Jul 15 14:48 faceattr
4.0K drwxr-xr-x 2 root root 4.0K Jul 15 14:48 facedet
4.0K drwxr-xr-x 2 root root 4.0K Jul 15 14:48 facenorm
4.0K drwxr-xr-x 2 root root 4.0K Jul 15 14:48 pedattr
4.0K drwxr-xr-x 2 root root 4.0K Jul 15 14:48 pedet
4.0K drwxr-xr-x 2 root root 4.0K Jul 15 14:48 pedrec
```

```
$ ls -lash /opt/findface-multi/models/
total 44K
4.0K drwxr-xr-x 11 root root 4.0K Jul 17 13:37 .
4.0K drwxr-xr-x 6 root root 4.0K Jul 19 15:31 ..
4.0K drwxr-xr-x 2 root root 4.0K Jul 17 13:37 carattr
4.0K drwxr-xr-x 2 root root 4.0K Jul 17 13:37 carnorm
4.0K drwxr-xr-x 2 root root 4.0K Jul 17 13:36 carrec
4.0K drwxr-xr-x 2 root root 4.0K Jul 17 16:20 detector
4.0K drwxr-xr-x 2 root root 4.0K Jul 17 16:24 face
4.0K drwxr-xr-x 2 root root 4.0K Jul 17 16:24 faceattr
4.0K drwxr-xr-x 2 root root 4.0K Jul 17 13:37 facenorm
4.0K drwxr-xr-x 2 root root 4.0K Jul 17 13:37 pedattr
4.0K drwxr-xr-x 2 root root 4.0K Jul 17 13:37 pedrec
```

Важно: Настоятельно рекомендуется отключить автоматическое обновление Ubuntu, чтобы сохранить совместимость FindFace Multi со средой установки. В этом случае вы сможете обновлять ОС вручную, контролируя процесс обновления отдельных пакетов.

Для отключения автоматического обновления Ubuntu выполните следующие команды:

```
sudo apt-get remove unattended-upgrades
sudo systemctl stop apt-daily.timer
sudo systemctl disable apt-daily.timer
sudo systemctl disable apt-daily.service
sudo systemctl daemon-reload
```

Примечание: Вы можете дополнительно развернуть Видеомagniфон. См. *Пошаговое развертывание Видеомagniфона*.

2.6.2 Резервное копирование и восстановление FindFace Multi и его данных

Вы можете создать резервную копию FindFace Multi перед его удалением, чтобы позже восстановить продукт и его данные.

В этом разделе:

- *Резервное копирование FindFace Multi*
- *Восстановление FindFace Multi и его данных*

Резервное копирование FindFace Multi

Чтобы создать резервную копию вашего экземпляра FindFace Multi и его данных, выполните следующие команды:

```
sudo tar -cvzf ~/configs.tar.gz -C /opt/findface-multi/ configs
sudo tar -cvzf ~/data.tar.gz -C /opt/findface-multi/ data
```

Восстановление FindFace Multi и его данных

Чтобы восстановить FindFace Multi и его данные из резервной копии, выполните следующие действия:

1. Загрузите файл инсталлятора `findface-*.run`.
2. Поместите файл `.run` в любой каталог на сервере установки (например, `/home/username`).
3. Из данного каталога сделайте файл `.run` исполняемым.

Примечание: Вместо `findface-*` укажите актуальное имя файла.

```
chmod +x findface-*.run
```

4. Запустите файл `.run`.

```
sudo ./findface-*.run
```

5. Пройдите процесс установки, как описано [здесь](#).
6. Чтобы восстановить FindFace Multi, его данные и конфигурационные файлы из резервной копии, после завершения установки остановите все контейнеры FindFace Multi.

```
cd /opt/findface-multi
sudo docker-compose stop
```

7. Удалите новые конфигурационные файлы и данные, сформированные и созданные инсталлером, и восстановите их из резервной копии.

```
sudo rm -r /opt/findface-multi/configs/*
sudo tar -xvf ~/configs.tar.gz -C /opt/findface-multi/
sudo rm -r /opt/findface-multi/data/*
sudo tar -xvf ~/data.tar.gz -C /opt/findface-multi/
```

8. Перезапустите контейнеры FindFace Multi.

```
cd /opt/findface-multi
sudo docker-compose up -d
```

2.6.3 Изменение структуры базы данных векторов признаков

В некоторых случаях вам может потребоваться применить новую структурную схему к базе данных векторов признаков Tarantool, например, при обновлении до последней версии продукта или если необходимо усовершенствовать структуру базы данных, добавив в нее дополнительные параметры, расширенные метаданные лиц и т. д.

В этом разделе:

- *О структуре биометрической базы данных*
- *Изменение структуры*

О структуре биометрической базы данных

В FindFace Multi структура базы данных задается через файл `/opt/findface-multi/configs/findface-tarantool-server/tnt-schema.lua`.

Структура представляет собой набор пространств (`spaces`) и полей, каждое из которых описывается следующими параметрами:

- **id:** id поля;
- **name:** название поля, должно совпадать с названием соответствующего параметра объекта;
- **field_type:** тип данных (`unsigned|string|set[string]|set[unsigned]`);
- **default:** значение по умолчанию. Если значение по умолчанию больше `1e14 - 1`, то его следует записывать в виде строки, т. е. `“123123...”` вместо `123123...`

Используемый по умолчанию файл `tnt_schema.lua` доступен по [ссылке](#).

Изменение структуры

Для изменения структуры базы данных выполните следующие действия:

1. Создайте резервную копию базы данных векторов признаков на основе Tarantool в любой выбранной директории, например, /etc/ffmulti_dump.

```
sudo docker exec -it findface-multi-findface-sf-api-1 bash -c "mkdir ffmulti_dump;
↪ cd ffmulti_dump && /storage-api-dump -config /etc/findface-sf-api.ini"
sudo docker cp findface-multi-findface-sf-api-1:/ffmulti_dump /etc
```

2. Измените структуру базы данных, добавив новую схему в файл tnt_schema.lua.

```
sudo docker run --rm --network host --volume '/opt/findface-multi/configs/findface-
↪ multi-legacy/findface-multi-legacy.py:/etc/findface-security/config.py:ro' docker.
↪ int.ntl/ntech/multi/multi/legacy:ffmulti-2.0.0 make-tnt-schema | sudo tee /opt/
↪ findface-multi/configs/findface-tarantool-server/tnt-schema.lua
```

3. Перейдите в директорию с файлами конфигурации Tarantool /opt/findface-multi/configs/findface-tarantool-server/. Убедитесь, что каждый шард shard-00*.lua содержит команду dofile. Удостоверьтесь, что переменные replication и spaces определены в параметре FindFace.start, как показано в примере.

```
sudo vi /opt/findface-multi/configs/findface-tarantool-server/shard-00*.lua

dofile("/tnt_schema.lua")

-- host:port to bind, HTTP API
FindFace = require("FindFace")
FindFace.start("127.0.0.1", 8101, {
    license_ntls_server="127.0.0.1:3133",
    replication = replication_master,
    spaces = spaces
})
```

4. Удалите данные из каталогов, соответствующих активным шардам.

```
sudo rm /opt/findface-multi/data/findface-tarantool-server/shard-*/{index,snapshots,
↪ xlogs}/*
```

5. Перезапустите шарды findface-tarantool-server.

```
docker restart findface-multi-findface-tarantool-server-shard-001-1 findface-multi-
↪ findface-tarantool-server-shard-002-1 findface-multi-findface-tarantool-server-
↪ shard-003-1 findface-multi-findface-tarantool-server-shard-004-1 findface-multi-
↪ findface-tarantool-server-shard-005-1 findface-multi-findface-tarantool-server-
↪ shard-006-1 findface-multi-findface-tarantool-server-shard-007-1 findface-multi-
↪ findface-tarantool-server-shard-008-1
```

6. Восстановите базу данных Tarantool из резервной копии.

```
sudo docker exec -it findface-multi-findface-sf-api-1 bash -c 'cd ffmulti_dump &&
↪ for x in *.json; do /storage-api-restore -config /etc/findface-sf-api.ini < "$x";
↪ done;'
```

Важно: Если некоторые прежние поля отсутствуют в новой структуре базы данных, сначала потребуется вручную удалить соответствующие данные из резервной копии.

См.также:

Пользовательские метаданные в Tarantool

2.6.4 Проверка статуса компонентов

Проверьте статус контейнеров, если вы столкнулись с проблемой в системе, используя следующие команды:

```
docker ps
sudo docker container inspect <container_id>/<container_name>
sudo docker container stats <container_id>/<container_name>
```

2.6.5 Логирование

Просмотр логов — это первое, что нужно сделать, чтобы определить причину проблемы. По умолчанию процессы FindFace Multi регистрируются в [логах контейнеров Docker](#), доступ к которым можно получить, используя команды `docker logs` и `docker service logs`. Помимо этого Docker использует [драйвер логирования в формате JSON](#), который кэширует логи контейнеров в данный формат. Вы можете настроить Docker на использование другого драйвера логирования, выбрав из [множества доступных механизмов логирования](#).

Данный раздел посвящен настройке Docker на использование драйвера логирования `journald`, который отправляет логи контейнеров в журнал `systemd`. В этом случае записи журнала извлекаются с помощью команды `journalctl`, через `journal API` или команды `docker logs`. Следуя приведенным ниже инструкциям, вы также можете настроить журнал `systemd`.

В этом разделе:

- *Настройка Journald*
- *Активация драйвера логирования journald*
- *Просмотр логов*

Настройка Journald

Для настройки сервиса `systemd-journal` выполните следующие действия:

1. Проверьте наличие директории `/var/log/journal`. Если таковая отсутствует, создайте ее следующей командой:

```
sudo mkdir /var/log/journal
sudo chmod 2755 /var/log/journal
```

- Откройте файл конфигурации `/etc/systemd/journald.conf`. Включите сохранение логов `journald` на жестком диске, раскомментировав параметр `Storage` и изменив его значение на `persistent`. Также отключите фильтрацию `systemd-journal`:

```
sudo vi /etc/systemd/journald.conf

[Journal]
...
Storage=persistent
...
RateLimitInterval=0
RateLimitBurst=0
...
```

При необходимости раскомментируйте и измените значение параметра `SystemMaxUse`. Данный параметр определяет максимальный объем логов на жестком диске. Укажите его значение в байтах или используйте К, М, Г, Т, Р, Е в качестве единиц для указанного размера (соответственно 1024, 1024², ... байт).

```
...
SystemMaxUse=3G
```

- Перезапустите сервис `journald`.

```
sudo systemctl restart systemd-journald.service
```

Активация драйвера логирования `journald`

Для того чтобы разрешить Docker использовать драйвер логирования `journald`, выполните следующие действия:

- Добавьте следующую строку в файл конфигурации `/etc/docker/daemon.json`.

Совет: Данный файл может отсутствовать в вашей системе. Проверьте, существует ли он, и если нет, сначала создайте каталог `/etc/docker`, а затем файл.

```
sudo mkdir /etc/docker
sudo touch /etc/docker/daemon.json
```

```
sudo vi /etc/docker/daemon.json

{
  "log-driver": "journald"
}
```

- Остановите все контейнеры Docker.

```
sudo docker stop $(sudo docker ps -a -q)
```

- Перезапустите сервис Docker.

```
sudo systemctl restart docker
```

4. Запустите все контейнеры Docker.

```
sudo docker start $(sudo docker ps -a -q)
```

Просмотр логов

Используйте любой удобный способ работы с логами `journald`. Мы рекомендуем начать со следующих команд:

- Отображение всех логов:

```
journalctl -fa
```

- Отображение логов по ID контейнера:

```
journalctl CONTAINER_ID=<container_id> -f
```

- Отображение логов по имени контейнера:

```
journalctl CONTAINER_NAME=<container-name> -f
```

См.также:

Журнал действий пользователей

2.6.6 Устранение неполадок с лицензированием и `findface-ntls`

При устранении неполадок с лицензией и сервером `findface-ntls` (см. *Лицензирование*) первым шагом является получение информации о лицензии и статусе сервера. Это можно сделать, отправив API-запрос в `findface-ntls`. Действия по устранению неполадок предпринимаются с учетом содержания API-ответа.

Совет: По вопросам устранения неполадок обращайтесь к нашим специалистам по адресу support@ntechlab.com.

Примечание: Онлайн-лицензирование выполняется через Глобальный менеджер лицензий NtechLab license.ntechlab.com. Проверьте его доступность. Для онлайн-лицензирования необходимы стабильное интернет-соединение и DNS.

Для получения информации о *лицензии* FindFace и статусе `findface-ntls`, выполните в консоли сервера `findface-ntls` следующую команду:

```
curl http://localhost:3185/license.json -s | jq
```

Ответ будет возвращен в формате JSON. Одним из наиболее значимых параметров в ответе является `last_updated`. Он показывает в секундах, как давно в последний раз проверялась локальная лицензия.

Интерпретируйте значение параметра `last_updated` следующим образом:

- [0, 5] — все работает отлично.
- (5, 30] — возможно имеют место быть какие-то проблемы со связью, либо с локальным накопителем, где хранятся файлы лицензий.

- (30; 120] — почти наверняка случилось что-то нехорошее.
- (120; ∞) — не удастся получить ответ от источника лицензирования в течение длительного времени. Необходимо вмешательство.
- "valid" -> "value": false: связь с источником лицензирования так и не была установлена.

```
curl http://localhost:3185/license.json -s | jq
{
  "name": "NTLS",
  "time": 1565186356,
  "type": "online",
  "license_id": "61063ce4b86945e1b70c3bdbedea453b",
  "generated": 1514467939,
  "last_updated": 5,
  "valid": {
    "value": true,
    "description": ""
  },
  "source": "/opt/ntech/license/import_
↪b68d7b7ec9a7310d18832035318cff0c9ddf11e3a9ab0ae962fbe48645e196d1.lic",
  "limits": [
    {
      "type": "time",
      "name": "end",
      "value": 1609161621
    },
    {
      "type": "number",
      "name": "faces",
      "value": 9007199254740991,
      "current": 0
    },
    {
      "type": "number",
      "name": "cameras",
      "value": 4294967295,
      "current": 0
    },
    {
      "type": "number",
      "name": "extraction_api",
      "value": 256,
      "current": 0
    },
    {
      "type": "boolean",
      "name": "gender",
      "value": true
    },
    {
      "type": "boolean",
      "name": "age",
      "value": true
    }
  ]
}
```

(continues on next page)

(продолжение с предыдущей страницы)

```
    },
    {
      "type": "boolean",
      "name": "emotions",
      "value": true
    },
    {
      "type": "boolean",
      "name": "fast-index",
      "value": true
    },
    {
      "type": "boolean",
      "name": "sec-genetec",
      "value": false
    },
    {
      "type": "boolean",
      "name": "beard",
      "value": false
    },
    {
      "type": "boolean",
      "name": "glasses",
      "value": false
    },
    {
      "type": "boolean",
      "name": "liveness",
      "value": false
    }
  ],
  "services": [
    {
      "name": "video-worker",
      "ip": "127.0.0.1:53276"
    },
    {
      "name": "FindFace-tarantool",
      "ip": "127.0.0.1:53284"
    },
    {
      "name": "FindFace-tarantool",
      "ip": "127.0.0.1:53288"
    }
  ]
}
```

2.6.7 Ручная очистка базы данных от старых данных

Совет: Для настройки автоматического удаления старых данных см. *Основные настройки*.

Для ручного удаления старых данных из базы данных FindFace Multi используйте утилиту `cleanup`. Вы можете выборочно удалить следующие данные:

- события с совпадениями (лица, силуэты, транспортные средства) и связанные эпизоды,
- события без совпадений (лица, силуэты, транспортные средства) и связанные эпизоды,
- полные кадры событий с совпадениями (лица, силуэты, транспортные средства),
- полные кадры событий без совпадений (лица, силуэты, транспортные средства),
- скриншоты счетчиков,
- кластерные события (лица, силуэты, транспортные средства),
- журнал действий пользователей,
- события внешних VMS.

Утилита очистки работает в контейнере `findface-multi-findface-multi-legacy-1`. Для того чтобы вызвать справку `cleanup`, выполните следующую команду:

```
sudo docker exec -it findface-multi-findface-multi-legacy-1 /opt/findface-security/bin/
python3 /tigre_prototype/manage.py cleanup --help

usage: manage.py cleanup [-h] [--as-configured]
                        [--face-events-max-fullframe-unmatched-age FACE_EVENTS_MAX_
↳FULLFRAME_UNMATCHED_AGE]
                        [--face-events-max-fullframe-matched-age FACE_EVENTS_MAX_
↳FULLFRAME_MATCHED_AGE]
                        [--face-events-max-unmatched-age FACE_EVENTS_MAX_UNMATCHED_AGE]
                        [--face-events-max-matched-age FACE_EVENTS_MAX_MATCHED_AGE]
                        [--body-events-max-fullframe-unmatched-age BODY_EVENTS_MAX_
↳FULLFRAME_UNMATCHED_AGE]
                        [--body-events-max-fullframe-matched-age BODY_EVENTS_MAX_
↳FULLFRAME_MATCHED_AGE]
                        [--body-events-max-unmatched-age BODY_EVENTS_MAX_UNMATCHED_AGE]
                        [--body-events-max-matched-age BODY_EVENTS_MAX_MATCHED_AGE]
                        [--car-events-max-fullframe-unmatched-age CAR_EVENTS_MAX_
↳FULLFRAME_UNMATCHED_AGE]
                        [--car-events-max-fullframe-matched-age CAR_EVENTS_MAX_
↳FULLFRAME_MATCHED_AGE]
                        [--car-events-max-unmatched-age CAR_EVENTS_MAX_UNMATCHED_AGE]
                        [--car-events-max-matched-age CAR_EVENTS_MAX_MATCHED_AGE]
                        [--car-cluster-events-max-age CAR_CLUSTER_EVENTS_MAX_AGE]
                        [--body-cluster-events-max-age BODY_CLUSTER_EVENTS_MAX_AGE]
                        [--face-cluster-events-max-age FACE_CLUSTER_EVENTS_MAX_AGE]
                        [--car-cluster-events-keep-best-max-age CAR_CLUSTER_EVENTS_KEEP_
↳BEST_MAX_AGE]
                        [--body-cluster-events-keep-best-max-age BODY_CLUSTER_EVENTS_
↳KEEP_BEST_MAX_AGE]
                        [--face-cluster-events-keep-best-max-age FACE_CLUSTER_EVENTS_
```

(continues on next page)

(продолжение с предыдущей страницы)

```

↪KEEP_BEST_MAX_AGE]
    [--area-activations-max-age AREA_ACTIVATIONS_MAX_AGE]
    [--audit-logs-max-age AUDIT_LOGS_MAX_AGE]
    [--counter-records-max-age COUNTER_RECORDS_MAX_AGE]
    [--external-vms-events-max-age EXTERNAL_VMS_EVENTS_MAX_AGE]
    [--external-vms-send-events-status-max-age EXTERNAL_VMS_SEND_
↪EVENTS_STATUS_MAX_AGE]
    [--remote-monitoring-events-max-age REMOTE_MONITORING_EVENTS_
↪MAX_AGE]
    [--configuration CONFIGURATION] [--version]
    [-v {0,1,2,3}] [--settings SETTINGS]
    [--pythonpath PYTHONPATH] [--traceback] [--no-color]
    [--force-color] [--skip-checks]

```

Delete FFSecurity entities

optional arguments:

```

-h, --help            show this help message and exit
--as-configured        Apply config age options for events, counter records
                        and clusters. Can't be used with other arguments.
--face-events-max-fullframe-unmatched-age FACE_EVENTS_MAX_FULLFRAME_UNMATCHED_AGE
                        face events max fullframe unmatched age to clean up
                        (in seconds)
--face-events-max-fullframe-matched-age FACE_EVENTS_MAX_FULLFRAME_MATCHED_AGE
                        face events max fullframe matched age to clean up (in
                        seconds)
--face-events-max-unmatched-age FACE_EVENTS_MAX_UNMATCHED_AGE
                        face events max unmatched age to clean up (in seconds)
--face-events-max-matched-age FACE_EVENTS_MAX_MATCHED_AGE
                        face events max matched age to clean up (in seconds)
--body-events-max-fullframe-unmatched-age BODY_EVENTS_MAX_FULLFRAME_UNMATCHED_AGE
                        body events max fullframe unmatched age to clean up
                        (in seconds)
--body-events-max-fullframe-matched-age BODY_EVENTS_MAX_FULLFRAME_MATCHED_AGE
                        body events max fullframe matched age to clean up (in
                        seconds)
--body-events-max-unmatched-age BODY_EVENTS_MAX_UNMATCHED_AGE
                        body events max unmatched age to clean up (in seconds)
--body-events-max-matched-age BODY_EVENTS_MAX_MATCHED_AGE
                        body events max matched age to clean up (in seconds)
--car-events-max-fullframe-unmatched-age CAR_EVENTS_MAX_FULLFRAME_UNMATCHED_AGE
                        car events max fullframe unmatched age to clean up (in
                        seconds)
--car-events-max-fullframe-matched-age CAR_EVENTS_MAX_FULLFRAME_MATCHED_AGE
                        car events max fullframe matched age to clean up (in
                        seconds)
--car-events-max-unmatched-age CAR_EVENTS_MAX_UNMATCHED_AGE
                        car events max unmatched age to clean up (in seconds)
--car-events-max-matched-age CAR_EVENTS_MAX_MATCHED_AGE
                        car events max matched age to clean up (in seconds)
--car-cluster-events-max-age CAR_CLUSTER_EVENTS_MAX_AGE
                        car cluster events max age to clean up (in seconds)

```

(continues on next page)

(продолжение с предыдущей страницы)

```

--body-cluster-events-max-age BODY_CLUSTER_EVENTS_MAX_AGE
    body cluster events max age to clean up (in seconds)
--face-cluster-events-max-age FACE_CLUSTER_EVENTS_MAX_AGE
    face cluster events max age to clean up (in seconds)
--car-cluster-events-keep-best-max-age CAR_CLUSTER_EVENTS_KEEP_BEST_MAX_AGE
    car cluster events keep best max age to clean up (in
seconds)
--body-cluster-events-keep-best-max-age BODY_CLUSTER_EVENTS_KEEP_BEST_MAX_AGE
    body cluster events keep best max age to clean up (in
seconds)
--face-cluster-events-keep-best-max-age FACE_CLUSTER_EVENTS_KEEP_BEST_MAX_AGE
    face cluster events keep best max age to clean up (in
seconds)
--area-activations-max-age AREA_ACTIVATIONS_MAX_AGE
    area activations max age to clean up (in seconds)
--audit-logs-max-age AUDIT_LOGS_MAX_AGE
    audit logs max age to clean up (in seconds)
--counter-records-max-age COUNTER_RECORDS_MAX_AGE
    counter records max age to clean up (in seconds)
--external-vms-events-max-age EXTERNAL_VMS_EVENTS_MAX_AGE
    external vms events max age to clean up (in seconds)
--external-vms-send-events-status-max-age EXTERNAL_VMS_SEND_EVENTS_STATUS_MAX_AGE
    external vms send events status max age to clean up
(in seconds)
--remote-monitoring-events-max-age REMOTE_MONITORING_EVENTS_MAX_AGE
    remote monitoring events max age to clean up (in
seconds)
--configuration CONFIGURATION
    The name of the configuration class to load, e.g.
    "Development". If this isn't provided, the
    DJANGO_CONFIGURATION environment variable will be
    used.
--version
    show program's version number and exit
-v {0,1,2,3}, --verbosity {0,1,2,3}
    Verbosity level; 0=minimal output, 1=normal output,
    2=verbose output, 3=very verbose output
--settings SETTINGS
    The Python path to a settings module, e.g.
    "myproject.settings.main". If this isn't provided, the
    DJANGO_SETTINGS_MODULE environment variable will be
    used.
--pythonpath PYTHONPATH
    A directory to add to the Python path, e.g.
    "/home/djangoprojects/myproject".
--traceback
    Raise on CommandError exceptions
--no-color
    Don't colorize the command output.
--force-color
    Force colorization of the command output.
--skip-checks
    Skip system checks.

```

Для полного удаления событий и эпизодов старше определенного количества дней используйте аргументы `--events-max-matched-age/--events-max-unmatched-age`. Например, для удаления событий с автомобилями без совпадений старше 5 дней (432000 секунд) выполните команду:


```
sudo docker exec -it findface-multi-findface-multi-legacy-1 /opt/findface-security/bin/
↳python3 /tigre_prototype/manage.py cleanup --car-events-max-unmatched-age 432000
```

Для удаления событий распознавания транспортных средств с совпадениями старше 5 дней (432000 секунд) выполните команду:

```
sudo docker exec -it findface-multi-findface-multi-legacy-1 /opt/findface-security/bin/
↳python3 /tigre_prototype/manage.py cleanup --car-events-max-matched-age 432000
```

Следующие команды удаляют только полные кадры событий распознавания силуэтов с совпадениями и без:

```
sudo docker exec -it findface-multi-findface-multi-legacy-1 /opt/findface-security/bin/
↳python3 /tigre_prototype/manage.py cleanup --body-events-max-fullframe-matched-age
↳432000
sudo docker exec -it findface-multi-findface-multi-legacy-1 /opt/findface-security/bin/
↳python3 /tigre_prototype/manage.py cleanup --body-events-max-fullframe-unmatched-age
↳432000
```

Для того чтобы удалить только скриншоты счетчиков, выполните команду:

```
sudo docker exec -it findface-multi-findface-multi-legacy-1 /opt/findface-security/bin/
↳python3 /tigre_prototype/manage.py cleanup --counter-records-max-age 432000
```

Для того чтобы удалить только кластерные события с лицами, выполните команду:

```
sudo docker exec -it findface-multi-findface-multi-legacy-1 /opt/findface-security/bin/
↳python3 /tigre_prototype/manage.py cleanup --face-cluster-events-max-age 432000
```

Для того чтобы удалить только записи в журнале действий пользователей, выполните команду:

```
sudo docker exec -it findface-multi-findface-multi-legacy-1 /opt/findface-security/bin/
↳python3 /tigre_prototype/manage.py cleanup --audit-logs-max-age 432000
```

Важно: Должен быть задан хотя бы один аргумент из указанных.

2.6.8 Сброс пароля

Для того чтобы сбросить пользовательский пароль в веб-интерфейсе FindFace Multi, выполните следующую команду:

```
sudo docker exec -it findface-multi-findface-multi-identity-provider-1 /opt/findface-
security/bin/python3 /tigre_prototype/manage.py changepassword <username>
```

2.6.9 Подключение S3-совместимого хранилища

S3 (Simple Storage Service) обеспечивает надежное и долговременное хранение неограниченного количества файлов и данных. Это позволяет избежать ограничений файловой системы при работе с большими объемами данных.

Настройка S3-совместимого хранилища

В этом разделе мы опишем как настроить S3-совместимое хранилище [MinIO](#) для работы с FindFace Multi. Если вы предпочитаете другое S3-совместимое хранилище, предполагается, что его настройку вы выполните самостоятельно.

1. *Подготовьте сервер на Ubuntu.*

Примечание: Работа MinIO была протестирована только на Ubuntu. Для получения информации о поддержке других ОС обратитесь к официальной документации [MinIO](#).

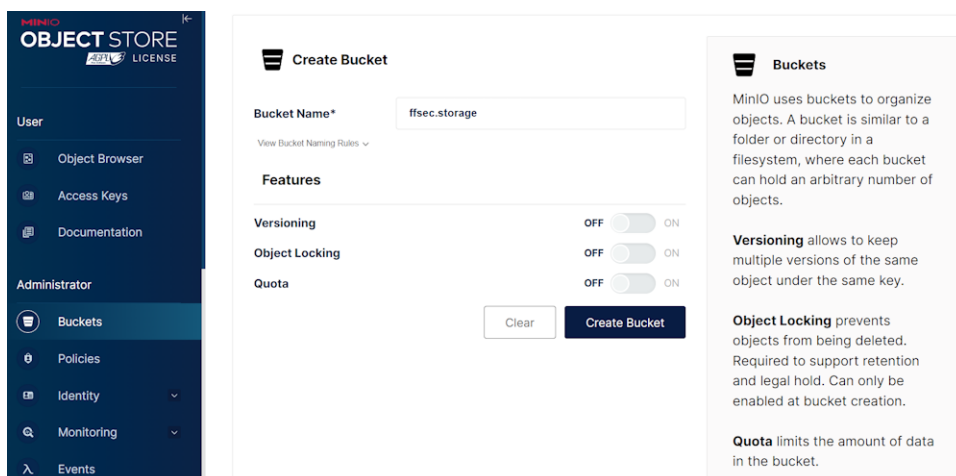
2. Чтобы запустить контейнер с MinIO, выполните следующую команду, задав логин/пароль в параметрах MINIO_ROOT_USER и MINIO_ROOT_PASSWORD:

```
docker run -d \  
-p 9003:9000 \  
-p 9004:9001 \  
-e "MINIO_ROOT_USER=admin" \  
-e "MINIO_ROOT_PASSWORD=admin_admin" \  
-v /path/to/storage_directory:/data \  
quay.io/minio/minio server /data --console-address ":9001"
```

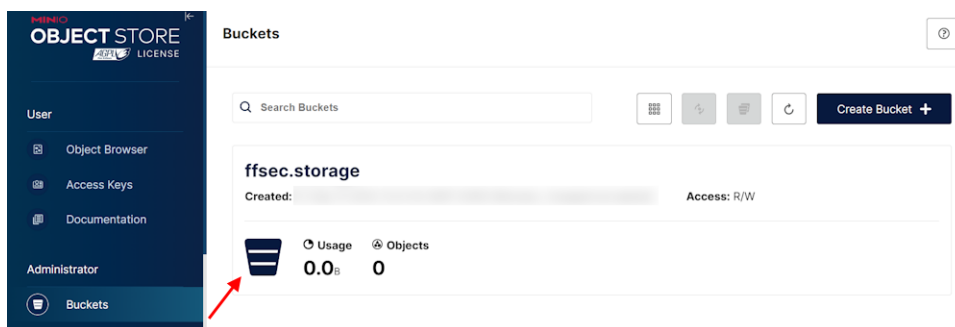
где /path/to/storage_directory – директория, выделенная для S3-совместимого хранилища.

3. Адрес веб-интерфейса MinIO будет зависеть от того, какой интерфейс вы выбрали при *установке* FindFace Multi на шаге #4.5. Используемый порт – 9004, например, 192.168.112.254:9004. Войдите в веб-интерфейс MinIO с учетными данными MINIO_ROOT_USER, MINIO_ROOT_PASSWORD, которые вы указали на предыдущем шаге.
4. В пользовательском интерфейсе MinIO создайте хранилище (bucket) (*Buckets -> Create Bucket*). Присвойте ему имя по умолчанию `ffsec.storage`.

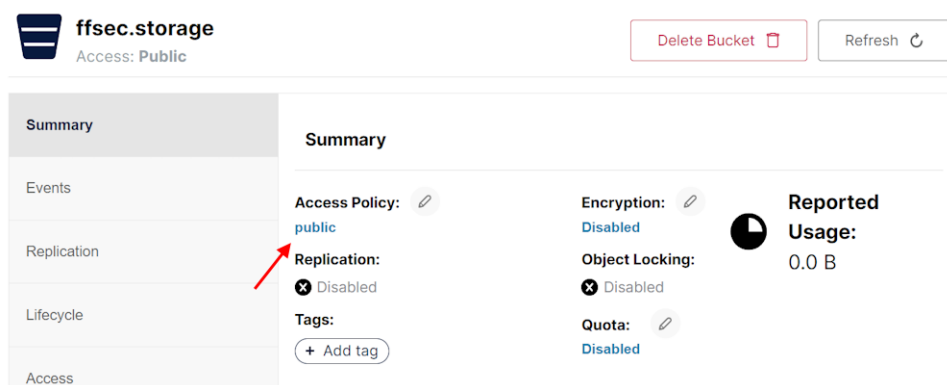
Примечание: Мы советуем оставить название хранилища `ffsec.storage`, предлагаемое по умолчанию, поскольку оно соответствует параметру `STORAGE_BUCKET_NAME` в файлах конфигурации `/opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py` и `/opt/findface-multi/configs/findface-multi-legacy/findface-multi-identity-provider.py`. Если вы используете другое имя для хранилища, не забудьте соответствующим образом изменить его в указанных выше файлах конфигурации.



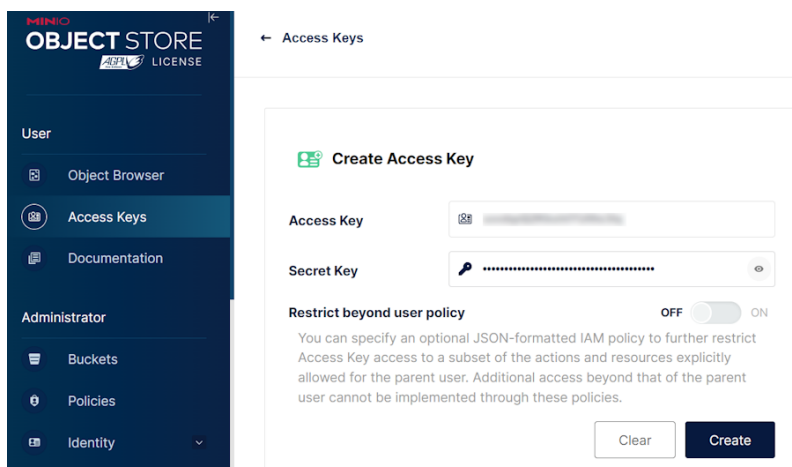
5. Щелкните на хранилище (bucket), чтобы открыть его настройки.



6. Измените доступ к хранилищу на public для корректной обработки видеоархивов.



7. В разделе *User -> Access Keys* создайте ключ доступа. Запишите значения Access key и Secret key, поскольку они понадобятся вам позже для настройки FindFace Multi.



Примечание: Интерфейс MinIO может отличаться. В более ранних версиях настройка ключа доступа производится в разделе *Identity* -> *Service Accounts* -> *Create service account*.

Настройка FindFace Multi для работы S3-совместимого хранилища

- В файле конфигурации `/opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py` перейдите в секцию `EXTERNAL_STORAGE_CONFIG`.

```
sudo vi /opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py

EXTERNAL_STORAGE_CONFIG = {
    'STORAGE_TYPE': None,
    'STORAGE_BUCKET_NAME': 'ffsec.storage',
    'ACCESS_KEY_ID': 'access-key',
    'SECRET_KEY': 'secret-key',
    'REGION_NAME': 'eu-west-3',
    'LOCAL_S3_SETTINGS': {
        'CONNECTION_ADDRESS': 'localhost:9003',
    },
}
```

Настройте параметры `EXTERNAL_STORAGE_CONFIG`:

- Параметр `STORAGE_TYPE` может принимать значения `'LOCAL'`, `'AWS'`, `None`:
 - `None` – локальная файловая система (по умолчанию)
 - `'LOCAL'` – любое S3-совместимое хранилище (например, MinIO)
 - `'AWS'` – хранилище Amazon S3.

Измените параметр `'STORAGE_TYPE'` на `'LOCAL'`, чтобы включить S3-совместимое хранилище.

- Значения параметров `'STORAGE_BUCKET_NAME'`, `'ACCESS_KEY_ID'` и `'SECRET_KEY'` должны соответствовать аналогичным значениям этих параметров, заданным на этапе инициализации MinIO.

3. 'REGION_NAME' – значение по умолчанию eu-west-3. Его можно изменить в настройках MinIO.
4. В 'LOCAL_S3_SETTINGS' -> 'CONNECTION_ADDRESS' измените localhost на IP адрес хоста MinIO, например, 192.168.112.254.
2. Таким же образом отредактируйте файл /opt/findface-multi/configs/findface-multi-identity-provider/findface-multi-identity-provider.py. Перейдите в секцию EXTERNAL_STORAGE_CONFIG и внесите изменения, аналогичные тем, которые были применены к файлу /opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py на предыдущем шаге.

```
sudo vi /opt/findface-multi/configs/findface-multi-identity-provider/findface-multi-identity-provider.py

EXTERNAL_STORAGE_CONFIG = {
    'STORAGE_TYPE': None,
    'STORAGE_BUCKET_NAME': 'ffsec.storage',
    'ACCESS_KEY_ID': 'access-key',
    'SECRET_KEY': 'secret-key',
    'REGION_NAME': 'eu-west-3',
    'LOCAL_S3_SETTINGS': {
        'CONNECTION_ADDRESS': 'localhost:9003',
    },
}
```

3. Для того чтобы пользовательский интерфейс мог загружать содержимое (миниатюры, полные кадры и т.д.), внесите изменения в файл /opt/findface-multi/configs/findface-multi-ui/nginx-site.conf. В секции server найдите параметр add_header Content-Security-Policy и добавьте адрес {ip_host_s3}:9003/, например:

```
sudo vi /opt/findface-multi/configs/findface-multi-ui/nginx-site.conf

...
server {
    listen 80 default_server;
    listen [::]:80 default_server;
    ...
    location / {
    ...
        add_header Content-Security-Policy "default-src 'self' 192.168.112.254:9003/
        ; img-src 'self' 192.168.112.254:9003/ blob: data:; media-src 'self' blob:;
        style-src 'self' 'unsafe-inline';";
    ...
}
```

4. Выполните пересборку контейнеров FindFace Multi из директории /opt/findface-multi.

```
cd /opt/findface-multi

sudo docker-compose down

sudo docker-compose up -d
```

5. Чтобы убедиться в том, что сервис FindFace Local S3 Storage запущен, изучите логи

```
findface-multi-findface-multi-legacy-1:
```

```
sudo docker logs findface-multi-findface-multi-legacy-1
```

Команда для переноса данных

Для переноса данных в удаленное или локальное хранилище используйте следующую команду:

```
sudo docker exec -it findface-multi-findface-multi-legacy-1 /opt/findface-security/bin/  
python3 /tigre_prototype/manage.py transfer_data --to_local / --to_remote
```

- Аргумент `--to_remote` переносит данные в удаленное хранилище.
- Аргумент `--to_local` переносит данные в локальную файловую систему.

Перед выполнением указанной команды убедитесь, что вы *настроили FindFace Multi* для использования S3-совместимого хранилища.

2.6.10 Перенос данных на другой диск

Высокая нагрузка на диск может приводить к задержкам при поступлении событий. В тяжелых случаях возможна полная неработоспособность FindFace Multi. Одним из способов снижения нагрузки на диск является перенос хранилищ данных FindFace Multi на другой диск.

В этом разделе:

- Подготовка диска
- Миграция хранилища фотографий и видеофрагментов
- Миграция основной базы данных (PostgreSQL)

Подготовка диска

Для того чтобы подготовить диск к миграции данных, выполните следующие действия:

1. Создайте новую точку монтирования (в примере `/mnt/ffdata`).

```
sudo mkdir /mnt/ffdata
```

2. Создайте раздел.

```
sudo parted /dev/sdb  
mklabel gpt  
mkpart primary ext4 1MiB 100%  
q  
sudo mkfs.ext4 /dev/sdb1
```

3. Узнайте UUID раздела (для `sdb1` в примере).

```
sudo blkid | grep sdb1
/dev/sdb1: LABEL="data" UUID="0638ebe0-853e-43ea-8f35-bfae305695d1" TYPE="ext4"
↳ PARTUUID="8cebaacc-77d7-4757-b4c6-14147e92646c"
```

4. Настройте автоматическое монтирование раздела при загрузке ОС, добавив запись в `fstab`.

```
sudo vi /etc/fstab
-----
#DATA mount
UUID=0638ebe0-853e-43ea-8f35-bfae305695d1 /mnt/ffdata/ ext4 auto,user,rw
↳ 0 2
-----
```

5. Смонтируйте все файловые системы.

```
sudo mount -a
```

Миграция хранилища фотографий и видеофрагментов

Примечание: Рекомендуем ознакомиться с разделом *Хранилища данных FindFace Multi* перед началом миграции хранилищ.

Для миграции хранилищ фотографий и видеофрагментов FindFace Multi выполните следующие действия:

1. Внутри точки монтирования создайте каталоги (например, `/mnt/ffdata/ffupload` и `/mnt/ffdata/findface-security`) для хранения фотографий и записанных видеофрагментов (если был развернут Видеомагнитофон). Затем переместите содержимое директорий `/opt/findface-multi/data/findface-upload/uploads/` и `/opt/findface-multi/data/findface-multi-legacy/uploads/` во вновь созданные каталоги.

```
sudo mkdir /mnt/ffdata/ffupload
sudo mkdir /mnt/ffdata/findface-security
sudo cp -ax /opt/findface-multi/data/findface-upload/uploads/ -R /mnt/ffdata/
↳ ffupload/
sudo cp -ax /opt/findface-multi/data/findface-multi-legacy/uploads/ -R /mnt/ffdata/
↳ findface-security/
sudo rm -r /opt/findface-multi/data/findface-multi-legacy/uploads/
sudo rm -r /opt/findface-multi/data/findface-upload/uploads/
```

2. Смонтируйте созданные каталоги (`/mnt/ffdata/ffupload` и `/mnt/ffdata/findface-security` в примере) в соответствующие контейнеры. Для этого откройте конфигурационный файл `/opt/findface-multi/docker-compose.yaml` и перечислите созданные каталоги в указанных в примере томах (volumes) разделов вместо каталогов по умолчанию.

```
sudo vi /opt/findface-multi/docker-compose.yaml

findface-upload:
    ...
    volumes: [ './configs/findface-upload/40-ffupload.sh:/docker-entrypoint.d/40-
↳ ffupload.sh:ro',
```

(continues on next page)

(продолжение с предыдущей страницы)

```

    '/mnt/ffdata/ffupload:/var/lib/ffupload']
...
findface-multi-identity-provider:
    ...
    volumes: ['./configs/findface-multi-identity-provider/findface-multi-identity-
    ↪provider.py:/etc/findface-security/config.py:ro',
    '/mnt/ffdata/findface-security/uploads:/var/lib/findface-security/uploads']
...
findface-multi-legacy:
    ...
    volumes: ['./configs/findface-multi-legacy/findface-multi-legacy.py:/etc/findface-
    ↪security/config.py:ro',
    '/mnt/ffdata/findface-security/uploads:/var/lib/findface-security/uploads']
...
findface-multi-ui:
    ...
    volumes: ['./configs/findface-multi-ui/nginx-site.conf:/etc/nginx/conf.d/default.
    ↪conf:ro',
    '/mnt/ffdata/findface-security/uploads:/var/lib/findface-security/uploads']

```

3. Выполните пересборку контейнеров FindFace Multi.

```

cd /opt/findface-multi

sudo docker-compose down

sudo docker-compose up -d

```

Миграция основной базы данных (PostgreSQL)

Для того чтобы мигрировать базу данных PostgreSQL, выполните следующие действия:

1. Остановите все контейнеры FindFace Multi:

```

cd /opt/findface-multi

sudo docker-compose down

```

2. Переместите каталог /opt/findface-multi/data/postgresql в каталог /mnt/ffdata.

```

sudo mv /opt/findface-multi/data/postgresql /mnt/ffdata

```

3. Укажите новый путь каталога postgresql (/mnt/ffdata/postgresql в примере) для монтирования в контейнер findface-multi-postgresql-1. Для этого в файле конфигурации /opt/findface-multi/docker-compose.yaml укажите его в томах (volumes) секции postgresql вместо каталога по умолчанию /opt/findface-multi/data/postgresql.

```

sudo vi /opt/findface-multi/docker-compose.yaml

postgresql:
    ...
    volumes: ['./configs/postgresql/40-init.sql:/docker-entrypoint-initdb.d/40-init.

```

(continues on next page)

(продолжение с предыдущей страницы)

```
↪sql:ro',
    '/mnt/ffdata/postgresql:/bitnami/postgresql']
```

4. Запустите все контейнеры FindFace Multi.

```
sudo docker-compose up -d
```

2.6.11 Отключение сервисов

Существует возможность отключить следующие сервисы FindFace Multi, если вы ими не пользуетесь:

- эпизоды
- менеджер очередей обработки видеоархивов
- вебхуки
- Active Directory
- очистка VMS

Для этого откройте файл конфигурации `/opt/findface-multi/config/findface-multi-legacy/findface-multi-legacy.py` и измените секцию `SERVICES`, установив `False` для сервисов, которые не будут использоваться. Перезапустите контейнер `findface-multi-findface-multi-legacy-1`.

```
sudo vi /opt/findface-multi/config/findface-multi-legacy/findface-multi-legacy.py

# disable unused services to increase
# overall system performance in some cases.
SERVICES = {
    "ffsecurity": {
        "episodes": False,
        "webhooks": False,
        # use queue manager to prevent drops of video archive events
        "video_archive_events_manager": False,
        "active_directory": False,
        "vms_cleanup": False,
    }
}
```

```
sudo docker container restart findface-multi-findface-multi-legacy-1
```

В результате соответствующие разделы исчезнут из веб-интерфейса.

Примечание: Раздел останется, если в нем есть какие-либо объекты (например, на вкладке *Вебхуки* созданы вебхуки). Однако сервис перестанет работать и генерировать данные.

2.6.12 Скрытие пунктов меню

Для того чтобы скрыть определенные пункты меню, выполните следующие действия:

1. Откройте файл конфигурации `/opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py`.

```
sudo vi /opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py
```

2. В разделе `FFSECURITY_UI_CONFIG` найдите секцию `menu`. Перечислите пункты меню, которые вы хотите скрыть.

```
FFSECURITY_UI_CONFIG = {
    "menu": {
        "disabled_items": [
            "areas",
            "player",
            "cases"
        ],
    },
}
```

Вы можете скрыть следующие пункты:

Пункт меню	Настройка
Поиск	"search"
Анализ взаимодействий	"relations"
Видеостена	"video-wall"
Счетчики	"counters"
Источники видео	"data_sources"
Эпизоды и события	"events_episodes"
Кластеры	"clusters"
Анализ аудитории	"analytics"
Настройки	"settings"
Картотека	"cards"
Видеоплеер	"player"
Сравнение	"verify"
Журнал действий	"audit-logs"
Отчеты	"reports"
Пользователи	"users"
Сессии	"sessions"
Блокировки	"blocklistRecords"
Группы камер	"camera-groups"
Списки наблюдения	"watch-lists"
Лицензия	"license"
Внешние VMS	"external-vms"
Вебхуки	"hooks"
Документация	"documentation"
API-документация	"api_doc"
пункты меню, появляющиеся при активации специализированных плагинов	Свяжитесь с нашей технической поддержкой, чтобы узнать подробнее о плагинах в своей системе.

3. Перезапустите контейнер `findface-multi-findface-multi-legacy-1`.

```
sudo docker container restart findface-multi-findface-multi-legacy-1
```

2.7 Приложения

2.7.1 Хранилища данных FindFace Multi

В этом разделе:

- *Список хранилищ*
- *Галереи базы данных векторов признаков*

Список хранилищ

FindFace Multi использует следующие хранилища данных:

- База данных векторов признаков на основе Tarantool, в которой хранятся векторы признаков объектов и события распознавания: `/opt/findface-multi/data/findface-tarantool-server`
- Основная база данных системы на основе PostgreSQL, в которой хранятся внутренние данные системы, карточки и учетные записи пользователей: `/opt/findface-multi/data/postgresql`
- Каталог `/opt/findface-multi/data/findface-upload/uploads`, в котором хранятся загруженные в карточки фотографии, видеофайлы, полные кадры событий, а также миниатюры объектов.
- Каталог, в котором хранятся такие артефакты событий, как нормализованные изображения объектов `/opt/findface-multi/data/findface-multi-legacy/uploads`

Примечание: В данном каталоге также хранятся видеофрагменты, если в системе активирована функция *Видеомагнитофон*.

- (Только вместе с Видеомагнитофоном) Хранилище базы данных на основе MongoDB `/opt/findface-multi/data/mongodb/`, в котором хранится метаданные о видеофрагментах, включая их точное местоположение в хранилище `/opt/findface-multi/data/findface-multi-legacy/uploads`.

Галереи базы данных векторов признаков

В базе данных векторов признаков на основе Tarantool по умолчанию есть следующие галереи:

- `ffsec_body_events`: векторы признаков, извлеченные из обнаруженных на видео силуэтов.
- `ffsec_body_objects`: векторы признаков, извлеченные из загруженных в картотеку изображений силуэтов.
- `ffsec_body_clusters`: центроиды кластеров силуэтов.
- `ffsec_car_events`: векторы признаков, извлеченные из обнаруженных на видео транспортных средств.

- `ffsec_car_objects`: векторы признаков, извлеченные из загруженных в картотеку изображений транспортных средств.
- `ffsec_car_clusters`: центроиды кластеров транспортных средств.
- `ffsec_face_events`: векторы признаков, извлеченные из обнаруженных на видео лиц.
- `ffsec_face_objects`: векторы признаков, извлеченные из загруженных в картотеку изображений лиц.
- `ffsec_face_clusters`: центроиды кластеров лиц.
- `ffsec_user_face`: векторы признаков, извлеченные из фотографий пользователей FindFace Multi для аутентификации по лицу.

В этой главе описывается, как работать с веб-интерфейсом FindFace, включая его расширенные возможности. Данные сведения будут интересны администраторам, операторам и другим пользователям.

3.1 Приступая к работе

После успешного *развертывания и конфигурирования* FindFace Multi пришло время открыть *веб-интерфейс* и начать работу. В этой главе вы найдете рекомендуемую последовательность шагов, следуя которой вы сможете задействовать весь спектр функциональных возможностей FindFace Multi.

В этой главе:

- *Подготовка к работе*
- *Добавление камер*
- *Создание списков наблюдения и картотеки*
- *Мониторинг объектов*
- *Организация видеонаблюдения*
- *Подсчет людей и транспортных средств. Измерение расстояния между людьми*
- *Связанная с людьми аналитика*
- *Работа в штатном режиме*
- *Базовое обслуживание системы*
- *Больше возможностей*

3.1.1 Подготовка к работе

Выполните первичную настройку системы:

1. Войдите в систему.
2. Выберите язык и темную или светлую тему.

3.1.2 Добавление камер

1. *Создайте новую группу камер* или используйте созданную по умолчанию. Группа камер — это системная сущность, которая позволяет группировать камеры с учетом их физического расположения. К примеру, камеры, расположенные на входе в здание, можно объединить в одну группу и работать с ними как с единым целым.
2. *Добавьте камеры* в созданную группу и *проверьте их работу*.

Дополнительно:

1. Настройте систему на обработку видеопотоков с группы камер в месте их физического расположения. Это может быть актуально в распределенной архитектуре. *Подробнее.*
2. Рассмотрите возможность включения дедупликации событий, если сцены наблюдения камер в группе перекрываются. Данная функция позволяет исключить регистрацию одинаковых событий в пределах одной группы камер. *Подробнее.*

3.1.3 Создание списков наблюдения и картотеки

1. Создайте *новый список наблюдения* или используйте созданный по умолчанию. Список наблюдения — это системная сущность, с помощью которой выполняется классификация объектов (лиц, силуэтов, транспортных средств) по произвольным критериям, например, нежелательные посетители, розыск, VIP, персонал и т. д.
2. Загрузите карточки и добавьте их в созданный список наблюдения *вручную, пакетно через веб-интерфейс* или используя функцию *пакетной загрузки через консоль*.

Дополнительно:

Настройте содержимое карточки. Создайте дополнительные поля, вкладки и поисковые фильтры.

3.1.4 Мониторинг объектов

По умолчанию FindFace Multi отслеживает только незарегистрированные в системе объекты, т. е. объекты, для которых *отсутствуют совпадения с картотекой*. Для того чтобы включить отслеживание объектов из списка наблюдения, сделайте его *активным*. Вы также можете включить звуковое оповещение и сделать обязательным принятие вручную связанных со списком событий.

Дополнительно:

1. *Включите автоматическую кластеризацию объектов одного происхождения:* изображений лица/силуэта, принадлежащих одному и тому же человеку, изображений одного транспортного средства.
2. Включите поддержку законов, связанных с обработкой персональных данных (GDPR и аналогичные). *Подробнее.*

3.1.5 Организация видеонаблюдения

Настройте раскладку камер для базового видеонаблюдения.

Дополнительно:

Для расширения функции видеонаблюдения используйте *Видеомагнитофон*.

3.1.6 Подсчет людей и транспортных средств. Измерение расстояния между людьми

Настройте *счетчики* для подсчета лиц, силуэтов и транспортных средств на подключенных камерах. Вы также можете настроить счетчики на измерение расстояния между силуэтами. Возможности счетчиков можно использовать в различных ситуациях, например, для подсчета людей в очередях и зонах ожидания, мониторинга общественных собраний, предотвращения скопления людей, обеспечения соблюдения санитарных правил, обнаружение транспортных заторов и т. д.

3.1.7 Связанная с людьми аналитика

FindFace Multi предоставляет набор инструментов для анализа людей:

1. *Анализируйте* социальные взаимодействия. Изучите круг людей, с которыми человек контактировал.
2. *Анализируйте аудиторию* по количеству посетителей, их полу, среднему возрасту, наиболее часто посещаемым зонам и характеру посещений (первое посещение или возвращение).

3.1.8 Работа в штатном режиме

1. *Идентифицируйте обнаруженные на видеоизображении объекты*, проверяя их в реальном времени на совпадение с объектами из списков наблюдения. Работайте с историей событий, используя различные фильтры.
2. Работайте с *эпизодами*. Эпизод — это набор событий идентификации, в которых фигурируют объекты одного происхождения (изображения лица и силуэта, принадлежащие одному человеку, и изображения одного транспортного средства), обнаруженные в течение определенного периода времени. Поскольку информация о событиях отображается на вкладке *События* в произвольном порядке, обработка большого количества разнородных событий может быть делом затруднительным и неэффективным. С функцией Эпизоды, система использует искусственный интеллект для группировки входящих событий на основе времени обнаружения и схожести объектов. Это позволяет с легкостью обрабатывать разнородные события даже в больших количествах.
3. Ищите объекты в базе обнаруженных объектов и картотеке. *Подробнее.*

4. *Ищите в архивных видео* объекты из списков наблюдения.
5. *Сравнивайте объекты* вручную, чтобы проверить их на совпадение.
6. *Стройте* детализированные отчеты на события распознавания объектов, эпизоды, события поиска, кластеры, камеры, картотеку, аудиторию или журнал действий пользователей.

3.1.9 Базовое обслуживание системы

1. *Настройте* автоматическое удаление старых событий, эпизодов, полных кадров и других старых данных.
2. При необходимости вручную *удалите* старые данные.
3. Регулярно создавайте *резервную копию* базы данных.
4. Поддерживайте безопасность системы, *анализируя журнал действий пользователей*.

3.1.10 Больше возможностей

1. Настройте *вебхуки* для автоматической отправки уведомлений об определенных событиях, эпизодах и записях счетчика на заданный URL-адрес. При наступлении нужного события FindFace Multi отправит HTTP-запрос на URL-адрес, указанный в настройках вебхука. Вебхуки можно использовать для решения разнообразных задач, например, для уведомления пользователя об определенном событии, вызова определенных действий на целевом веб-сайте, при решении задач безопасности, таких как удаленное автоматическое управление доступом и др. *Подробнее*.
2. Задействуйте функции FindFace Multi через *HTTP API*.

См.также:

- *Управление пользователями и безопасность системы*
- *Источники видео*
- *Картотека*
- *Счетчики лиц, силуэтов и транспортных средств. Определение дистанции*
- *Кластеры лиц, силуэтов, транспортных средств*
- *Связанная с людьми аналитика*
- *Обслуживание и устранение неисправностей*

3.2 Основы веб-интерфейса

Работа с FindFace Multi выполняется через веб-интерфейс. Для того чтобы отобразить веб-интерфейс, в адресной строке браузера введите базовый адрес веб-интерфейса и пройдите авторизацию.

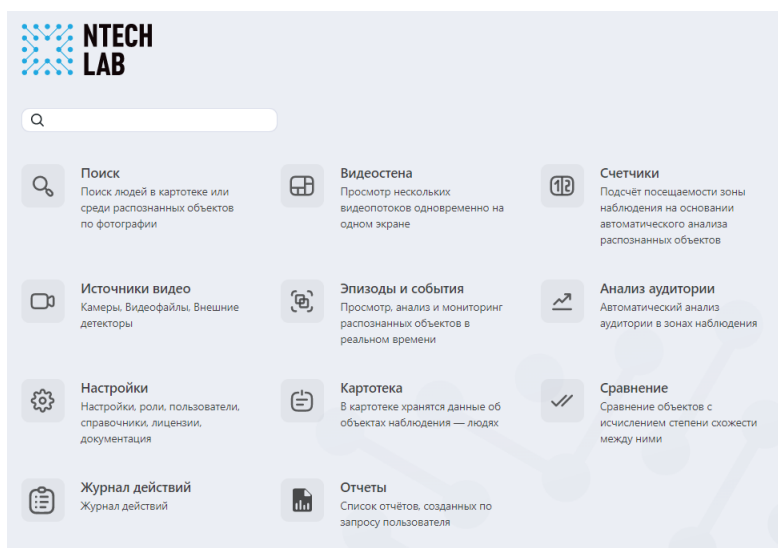
Примечание: Базовый адрес задается при *установке* FindFace Multi.

Важно: Для первого входа в систему после развертывания FindFace Multi используйте учетную запись администратора, созданную при *установке*. Для создания других пользователей см. раздел *Управление пользователями*.

Совет: Поднимите безопасность своей системы на новый уровень, используя *аутентификацию по лицу*.

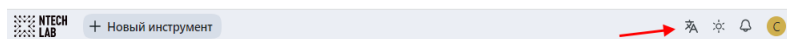
3.2.1 Навигация

В панели навигации разное количество элементов в зависимости от роли пользователя.



3.2.2 Язык веб-интерфейса и тема

Для переключения языка веб-интерфейса и выбора темы нажмите на  на верхней панели.



3.3 Картотека

В картотеке хранятся карточки лиц, содержащие изображения лиц, биометрические сведения и связанные документы, а также карточки транспортных средств, содержащие изображения транспортных средств, регистрационный номер и связанные документы.

Для одновременного создания большого количества карточек используйте функциональность *пакетной загрузки через консоль*.

В этом разделе:

- Создание карточки
- Пакетная загрузка карточек
- Фильтрация карточек
- Очистка картотеки

3.3.1 Создание карточки

Для создания карточки вручную выполните следующие действия:

1. Перейдите на вкладку *Картотека*.
2. Нажмите *+ Новая карточка*.



3. Введите название карточки. При необходимости добавьте комментарий.
4. Из раскрывающегося списка *Списки наблюдения* выберите список, в который нужно добавить карточку (или несколько списков, по очереди).
5. Поставьте флажок *Карточка активна*. Если карточка неактивна, она не будет использоваться для мониторинга.
6. Сохраните карточку. В результате в ней появятся дополнительные вкладки.
7. На той же вкладке *Сведения* прикрепите связанные файлы.

Mrs. Smith

Сведения Лица События кластеров лиц Силуэты События

Название карточки
Mrs. Smith

Списки наблюдения
Hitmen

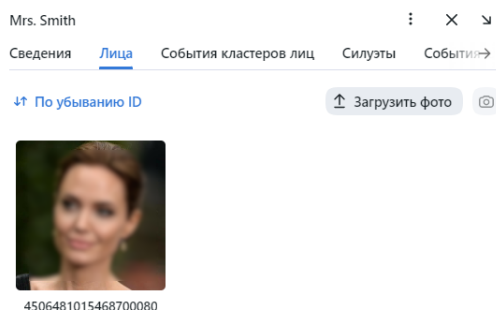
Комментарий

☒ Карточка активна

Нет файлов [Прикрепить первый](#)

ID 2
Создано 14.03.2023 14:56:32
Обновлено 14.03.2023 15:08:05

8. Для карточек «Люди» на вкладке *Лица* прикрепите фотографии лица человека, а на вкладке *Силуэты* прикрепите фотографии силуэта человека. Поддерживаемые форматы: WEBP, JPG, BMP, PNG.

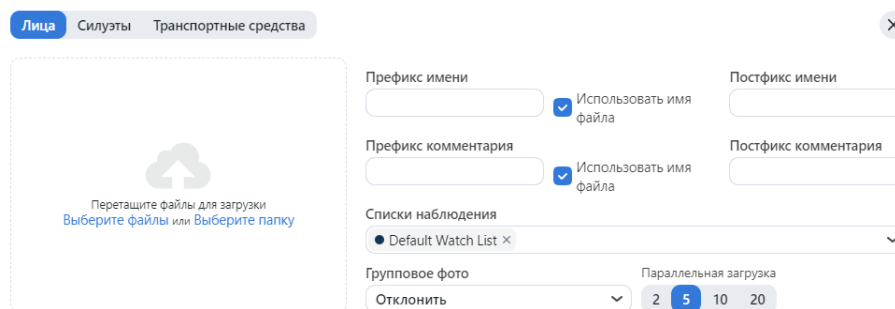


9. Для карточек «Транспортные средства» на вкладке *Транспортные средства* прикрепите соответствующие фотографии.

3.3.2 Пакетная загрузка карточек

Для создания большого количества карточек используйте функциональность пакетной загрузки. Выполните следующие действия:

1. Перейдите на вкладку *Картотека*.
2. Нажмите *+ Новая карточка* -> Попробуйте *пакетную загрузку карточек*.
3. Выберите файлы или папку и перетащите их для загрузки.



4. Введите имя. При необходимости добавьте комментарий.
5. Нажмите *Начать*.

3.3.3 Фильтрация карточек

Наиболее часто используемые фильтры для карточек отображаются в верхней части окна.

Для того чтобы отобразить набор доступных фильтров целиком, нажмите на кнопку *Все фильтры*. Набор фильтров следующий:

- *Тип карточки*: отобразить карточки по выбранному типу (люди или транспортные средства).
- *Списки наблюдения*: отобразить карточки, включенные в выбранные списки наблюдения.
- *Лица*: отфильтровать карточки по присутствию биометрических данных лица человека.

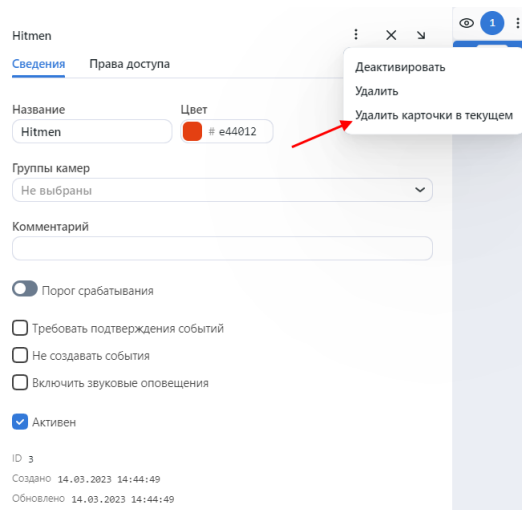
- *Силуэты*: отфильтровать карточки по присутствию биометрических данных силуэта человека.
- *Транспортные средства*: отфильтровать карточки по присутствию данных о транспортном средстве.
- *Регистрационный номер*: фильтровать карточки по регистрационному номеру.
- *Заполнение*: отобразить только пустые карточки, только заполненные или любые.
- *Название содержит*: фильтровать карточки по имени.
- *ID*: отобразить карточку с определенным ID.
- *Статус*: фильтровать карточки по статусу.

Вы можете отсортировать карточки в списке по параметру *ID*.

3.3.4 Очистка картотеки

Вы можете очистить картотеку целиком или по спискам наблюдения одним щелчком мыши. Выполните следующие действия:

1. Перейдите *Настройки* -> *Списки наблюдения*.
2. Выберите один или несколько списков наблюдения.
3. Нажмите *Удалить карточки в текущем*.



3.4 Источники видео

Для настройки видеомониторинга объектов добавьте камеры в FindFace Multi, сгруппировав их с учетом расположения.

Примечание: Права на создание групп камер и камер настраиваются в разрешениях пользователя (см. *Управление ролями и пользователями*).

В этой главе:

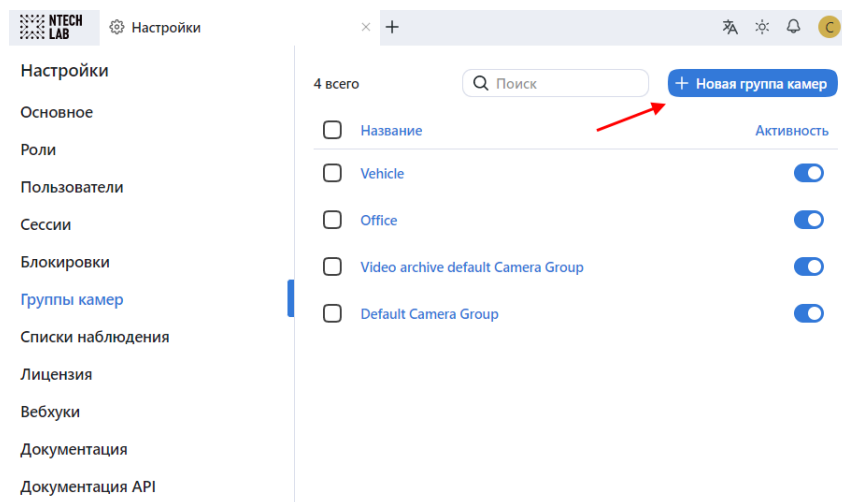
- Создание группы камер
- Добавление камеры
- Параметры обработки видео
- Загрузка и обработка видеофайла
- Внешние детекторы
- Мониторинг работы камер

3.4.1 Создание группы камер

Совет: В системе доступна группа камер по умолчанию.

Для создания группы камер выполните следующие действия:

1. Перейдите на вкладку *Настройки*. Выберите *Группы камер*.
2. Нажмите *+ Новая группа камер*.



3. На вкладке *Сведения*, введите имя группы и при необходимости комментарий к ней.

4. Если вам нужно выделить определенный экземпляр `findface-video-worker` для обработки видеопотоков с данной группы камер, создайте или выберите из уже созданных одну или несколько меток.

Примечание: Для того чтобы завершить выделение, перечислите метки в файле конфигурации `findface-video-worker.yaml`. Подробнее см. [Привязка группы камер к экземпляру findface-video-worker](#).

5. Если события от камер, принадлежащих одной группе, требуется дедуплицировать, т. е. исключить одинаковые события, поставьте флажок *Убрать дубли событий с интервалом* и задайте в секундах интервал дедупликации (интервал, с которым события проверяются на уникальность).

Предупреждение: Используйте дедупликацию очень осторожно. Если камеры из одной группы наблюдают разные сцены, некоторые объекты могут быть пропущены. Подробнее см. [Дедупликация событий](#).

6. По умолчанию на всех группах камер в системе используется *универсальный порог срабатывания*. Для того чтобы установить индивидуальный порог для группы камер, поставьте флажок *Порог срабатывания* и укажите нужное пороговое значение.

Важно: Для большинства случаев распознавания лиц значение, установленное по умолчанию для универсального порога срабатывания, является оптимальным. Не рекомендуется самостоятельно менять его. Обязательно предварительно проконсультируйтесь с нашими техническими специалистами (support@ntechlab.com).

7. Нажмите *Сохранить*.
8. Поставьте флажок *Активность*.
9. На вкладке *Права доступа* назначьте права на работу с группой камер, указав, пользователям с какими ролями разрешено изменять/просматривать ее настройки.

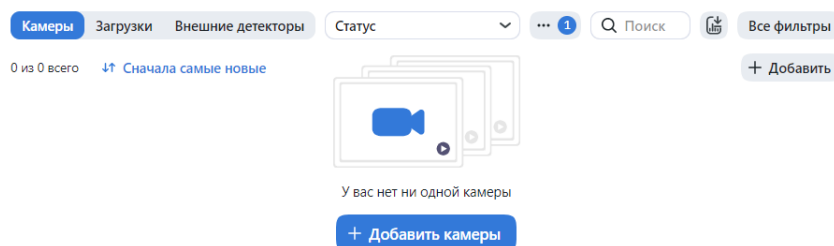
Office	:	X	↘
Сведения	<u>Права доступа</u>		
3 всего	Поиск		
Название	Просмотр	Изменить	
Administrator	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Operator	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
User	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

10. Нажмите *Сохранить*.

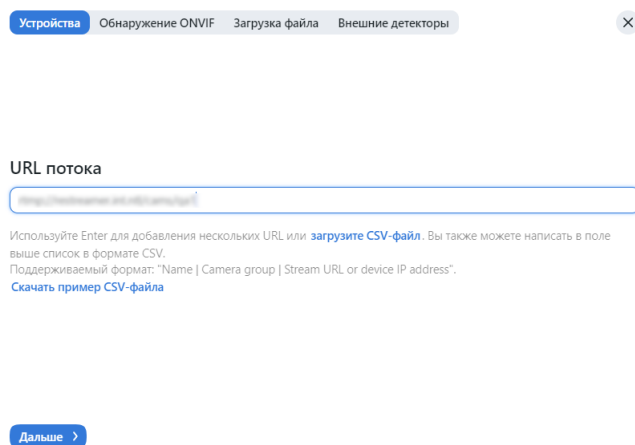
3.4.2 Добавление камеры

Для добавления камеры выполните следующие действия:

1. Перейдите на вкладку *Источники видео* -> *Камеры*.
2. Нажмите *+ Добавить*.



3. На вкладке *Устройства* введите URL потока и нажмите *Далее*.



- Укажите имя камеры.
- Добавьте камеру в группу.
- Поставьте флажки для тех детекторов, которые нужно активировать для данной камеры: лица, силуэты, транспортные средства.

← Назад ✕

Вы добавляете 1 устройство

Название	Группа камер	URL	Лица	Силуэты	Транспортные средства
<input type="text" value="Camera_Name"/>	<input type="text" value="Default Camera Group"/>	<input type="text" value="http://findface.com:8080/"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

- Нажмите *Добавить и настроить*. Вы увидите, как появятся дополнительные вкладки.
- Нажмите на камеру в списке, чтобы открыть мастер настройки обработки. Настройте *параметры обработки видео*.
 - Если вы используете камеру ONVIF, выберите ее из списка обнаруженных устройств для автоматической загрузки доступных настроек и видеопотоков.

Устройства **Обнаружение ONVIF** Загрузка файла Внешние детекторы

9 устройств [Найти устройства](#)


<input type="checkbox"/>	Бренд	Модель	IP-адрес	Порт
<input type="checkbox"/>			192.168.1.100	80
<input type="checkbox"/>			192.168.1.101	80
<input type="checkbox"/>			192.168.1.102	80
<input type="checkbox"/>			192.168.1.103	80
<input type="checkbox"/>			192.168.1.104	80
<input type="checkbox"/>			192.168.1.105	80
<input type="checkbox"/>			192.168.1.106	80
<input type="checkbox"/>			192.168.1.107	80
<input type="checkbox"/>			192.168.1.108	80
<input type="checkbox"/>	IPC		192.168.1.109	80

3.4.3 Параметры обработки видео

- Для каждой камеры предоставляется полная статистика: продолжительность текущей сессии обработки, количество отправленных объектов, количество объектов, обработанных с ошибками после последнего перезапуска job-задания, количество пропущенных кадров и другие данные. Для просмотра этих данных нажмите на камеру и перейдите на вкладку *Сведения*.

Camera 1 Не активна ▶ ⋮ ✕ ↵

Сведения Основные Дополнительные Зоны Лица Сил→



Сведения

ID	3
Статус	DISABLED
Длительность процесса	98.72
Потеряно кадров	2060
Перезапусков	1

Статистика по объектам

Отправлено лиц	69
Не отправлено лиц	0
Неудачных лиц	142
Отправлено силуэтов	186
Не отправлено силуэтов	0
Неудачных силуэтов	429
Отправлено транспортных средств	1
Не отправлено транспортных средств	0

2. На вкладке *Основные* вы можете изменить имя камеры, группу камер и описание. Вы можете записать видео и включить liveness. Поставьте флажки для тех детекторов, которые нужно активировать для данной камеры: лица, силуэты, транспортные средства.

Camera 1 Не активна ▶ ⋮ ✕ ↵

Сведения **Основные** Дополнительные Зоны Лица Сил→

Сведения

Название камеры: Группа камер:

Описание:

☒ Записывать видео

☒ Включить liveness

Детекторы

☒ Лица ☒ Силуэты ☒ Транспортные средства

Тип подключения


URL потока

3. На вкладке *Дополнительные* выполните тонкую настройку камеры:

Camera 1 Не активна ▶ ⋮ ✕ ↵

Сведения Основные **Дополнительные** Зоны Лица Сигн→

Трансформация



- ☐ Отразить по вертикали
- ☐ Отразить по горизонтали
- ☐ Повернуть по часовой стрелке
- ☐ Повернуть против часовой стрелки

Отправка лиц

Таймаут ⬇ ⬆ 15000 Мс

☒ Проверить SSL-сертификат

Временная метка

☐ Получать метки времени из потока Добавить к временным меткам ⬇ ⬆ 0 Сек

Другие

Параметры FFmpeg ⬅

Ограничение скорости воспроизведения ⬇ ⬆ -1 Формат ввода Минимальная интенсивность движения ⬇ ⬆ 0

☐ Считывание кадров из источника без пропусков

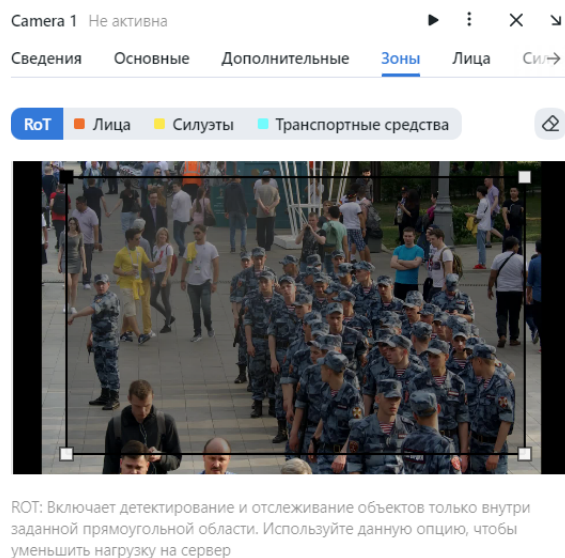
- При необходимости измените ориентацию видео.

Важно: Вращение потока выполняется на стороне сервера `findface-multi-legacy` средствами постобработки. Это может негативно сказаться на производительности. Рекомендуем по возможности делать вращение потока на стороне камеры.

- *Таймаут:* Время ожидания в миллисекундах отправки на сервер обнаруженных объектов.
- *Проверить SSL-сертификат:* Поставьте флажок, чтобы включить проверку подписи SSL-сертификата при взаимодействии по https видеодетектора объектов с сервером. Снимите флажок, если вы используете самоподписанный сертификат.
- *Получать метки времени из потока:* Поставьте флажок, чтобы отправлять на сервер временные метки, полученные из потока. Снимите флажок, чтобы отправлять текущие дату и время.
- *Добавить к временным меткам:* Прибавлять указанное количество секунд к временным меткам из потока.
- *Параметры FFmpeg:* Параметры FFmpeg для видеопотока. Задаются массивом строк `ключ=значение`, например, `["rtsp_transpotr=tcp", "ss=00:20:00"]`.
- *Ограничение скорости воспроизведения:* Если меньше нуля, то скорость не ограничивается, в остальных случаях поток читается с заданной скоростью `play_speed`. Не применимо для потоков с камер видеонаблюдения.
- *Формат ввода:* Передать формат FFmpeg (mxg, flv, и т. д.), если он не может быть автоматически определен.
- *Минимальная интенсивность движения:* Минимальная интенсивность движения, которая будет регистрироваться детектором движения.

- *Считывание кадров источника без пропусков*: По умолчанию (false), если findface-video-worker не обладает достаточными ресурсами для обработки всех кадров с объектами, он будет пропускать некоторые из них. Если данная опция активна (true), findface-video-worker помещает лишние кадры в очередь, чтобы обработать их впоследствии. При обработке файлов как камер необходимо указать значение true.

4. Укажите область отслеживания (*Зоны*) в поле зрения камеры и область интереса. Нажмите *Сохранить*.



Область отслеживания позволяет обнаруживать и отслеживать лица/силуэты и транспортные средства только внутри заданного прямоугольника. Используйте данную опцию, чтобы уменьшить нагрузку на видеодетектор объектов. Область интереса позволяет отправлять на сервер только объекты, обнаруженные в ее пределах.

5. На вкладках *Лица*, *Силуэты* and *ТС* укажите настройки детектора объектов каждого типа.

Камера 1 Не активна ▶ ⋮ ✕ ↵

Сведения Основные **Дополнительные** Зоны Лица Сил→

☒ Включен Сбросить параметры лиц

Фильтр _____

Размер - Качество

Сжатие _____

Качество JPEG

☐ Полный кадр PNG

Трекинг _____

Коэффициент перекрытия границ распознанного объекта

Максимальная длительность трека

Интервал закрытия трека

☐ Отправлять историю трека

ROT _____

☐ Обрезать полный кадр

Отправка на обработку _____

☒ Буферный режим

Интервал

☐ Отправлять первый распознанный объект немедленно ☐ Отправлять в каждом интервале

☐ Отправлять первый кадр трека ☐ Отправлять последний кадр трека

☒ Отправлять лучшее нормализованное изображение трека ☒ Отправлять лучший кадр трека

- **Размер:** Минимальный и максимальный размер отправляемых объектов, в пикселях.
- **Качество:** Минимальное пороговое значение качества изображения лица. Допустимый диапазон от 0 до 1. Базовое рекомендуемое значение 0,45 — соответствует удовлетворительному качеству изображения лица. Не меняйте значение по умолчанию без предварительной консультации с нашими техническими экспертами (support@ntechlab.com).
- **Качество JPEG:** Качество сжатия полного кадра для отправки.
- **Полный кадр PNG:** Отправлять полные кадры в формате PNG, а не в формате JPEG, как установлено по умолчанию. Не включайте этот параметр, не посоветовавшись с нашей командой, так как он может повлиять на работу всей системы.
- **Коэффициент перекрытия границ распознанного объекта:** Процент перекрытия границ между двумя последовательными кадрами, чтобы эти bbox'ы считались одним треком. Диапазон значений: от 0 до 1. Не меняйте значение по умолчанию без предварительной консультации с нашими техническими экспертами (support@ntechlab.com).
- **Максимальная длительность трека:** Максимальное приблизительное количество кадров в треке, по достижении которого трек будет принудительно закрыт. Включите опцию, чтобы автоматически завершать вечные треки, например, треки детектирования лиц с рекламных носителей.
- **Интервал закрытия трека:** Завершить трек, если в нем не было распознано новых изображений объекта в течение заданного времени (в секундах).
- **Отправить историю трека:** Отправлять массив координат границ объекта вместе с собы-

тием. Может быть применено для внешних интеграций, чтобы отобразить путь объекта.

- *Обрезать полный кадр*: Поставьте флажок если необходимо обрезать полный кадр по размеру ROT-области перед отправкой на распознавание. Размер полного кадра будет равен размеру ROT-области.
- *Буферный режим*: По умолчанию для обработки видео система использует буферный режим, т.е. отправляет по одному кадру наилучшего качества на трек для экономии места на диске. Отключите данный режим, чтобы получать больше кадров лица, если это необходимо. Если буферный режим включен, параметры режима реального времени не работают.

Параметры режима реального времени:

Примечание: Эти параметры не работают, если включен буферный режим.

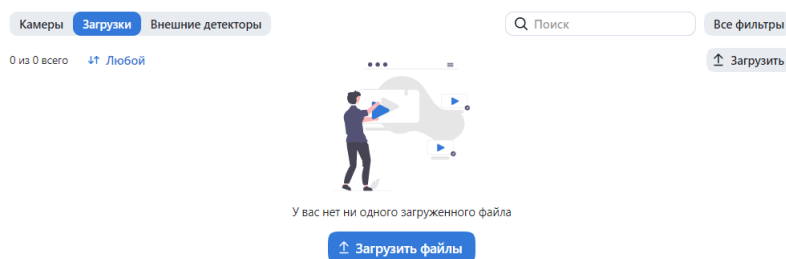
- *Интервал*: Временной интервал в секундах (целое или десятичное число), в течение которого в режиме реального времени выбирается лучший кадр с объектом.
- *Отправлять первый распознанный образ немедленно*: Поставьте флажок, чтобы отправлять первый объект из трека сразу, как только фильтры по качеству/размеру/региону интереса пройдены, не дожидаясь окончания первого `realtime_post_interval` в режиме реального времени. Снимите флажок, чтобы отправлять первый объект из трека только после окончания первого `realtime_post_interval`.
- *Отправлять первый кадр трека*: При завершении трека в дополнение к overall-кадру трека будет дополнительно отправлен первый кадр трека. Может быть применимо для внешних интеграций.
- *Отправлять лучшее нормализованное изображение*: Отправлять лучшие нормализованные изображения обнаруженных объектов.
- *Отправлять в каждом интервале*: Поставьте флажок, чтобы отправлять лучший кадр в каждом временном `realtime_post_interval` в режиме реального времени. Снимите флажок, чтобы отправлять лучший кадр, только если его качество улучшилось по сравнению с предыдущим отправленным кадром.
- *Отправлять последний кадр трека*: При завершении трека в дополнение к overall-кадру трека будет дополнительно отправлен последний кадр трека. Может быть применимо для внешних интеграций.
- *Отправлять лучший кадр трека*: Отправлять лучшие полные кадры с обнаруженными объектами.

6. (Опционально) На вкладке *Геопозиция* укажите географические координаты камеры.

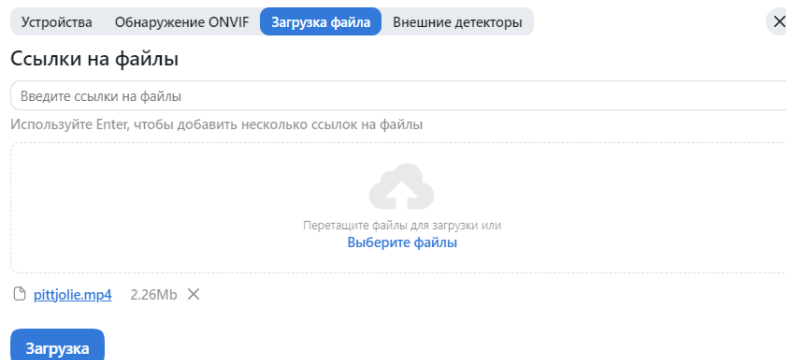
3.4.4 Загрузка и обработка видеофайла

Для того чтобы загрузить и обработать видеозапись, выполните следующие действия:

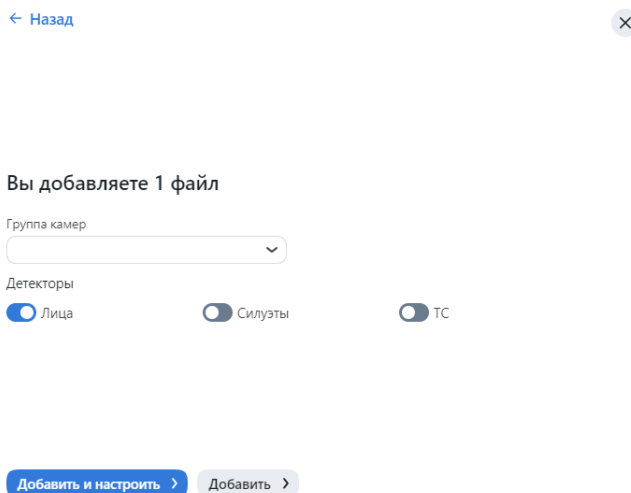
1. Перейдите на вкладку *Источники видео* -> *Загрузки*.
2. Нажмите *Загрузить*.



3. Укажите URL или выберите файл. Нажмите *Загрузка*.



4. Назначьте группу камер, к которой система будет приписывать события с объектами из данного видео. Группа камер **Video archive default Camera Group** идеально подходит для этой задачи. Вы также можете создать *новую группу камер* с базовыми настройками специально для данного видео. Поставьте флажки для тех детекторов, которые нужно активировать для данного видео: лица, силуэты, транспортные средства.



Нажмите *Добавить и настроить*. В результате видеозапись будет загружена и добавлена в список.

5. Нажмите на видео в списке, чтобы открыть мастер настройки обработки. Настройте *параметры обработки видео* аналогичным образом, как и для камер.

- (Опционально) Выберите камеру в составе группы, чтобы более точно промаркировать события распознавания объектов из этого видео.
 - (Опционально) Настройте метки времени для событий распознавания объектов.
6. Нажмите три точки -> *Обработать текущий*, чтобы запустить идентификацию объектов. Вы можете просмотреть события идентификации объектов на вкладках *События* и *Эпизоды*, отфильтровав список событий по группе камер/ камере, ассоциированной с видео.

3.4.5 Внешние детекторы

Вы также можете использовать объект типа камера для интеграции внешнего детектора.

При необходимости изменить имя внешнего детектора. Укажите исходную группу камер и описание. При необходимости, поставьте флажок *Включить liveness*.

test Не активна

test

Исходная группа камер

test

Default Camera Group

Описание

Включить liveness

Токен

ID 16

Создано 02.03.2023 21:42:22

Обновлено 02.03.2023 21:42:22

3.4.6 Мониторинг работы камер

Мониторинг работы камер выполняется на вкладке *Камеры*.

Камеры	Загрузки	Внешние детекторы	Статус	Поиск	Все фильтры
14 из 14 всего	Сначала самые новые	Добавить			
Изображение	Название	Статус			
<input type="checkbox"/>	MStreet 2 Street	● INPROGRESS			
<input type="checkbox"/>	Shopping Mall 1 Street	● INPROGRESS			
<input type="checkbox"/>	SD Default Camera Group	● DISABLED			

Статусы камер:

- Зеленый: идет обработка видеопотока с камеры, проблем не обнаружено.
- Желтый: камера работает менее 30 секунд или имеют место одна или несколько ошибок при отправке объектов.
- Красный: камера не работает.
- Серый: камера отключена.

Совет: Вы можете настроить отображение желтого и красного статусов на основании доли пропущенных кадров и доли неудачных отправок объектов на сервер. Для этого измените следующие параметры в файле конфигурации `/opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py`:

```
sudo vi /opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py

FFSECURITY = {
    ...

    # max camera frames_dropped percent
    'MAX_CAMERA_DROPPED_FRAMES': {'yellow': 0.1, 'red': 0.3},
    # max camera objects failed percent
    'MAX_CAMERA_FAILED_OBJECTS': {'yellow': 0.1, 'red': 0.3},

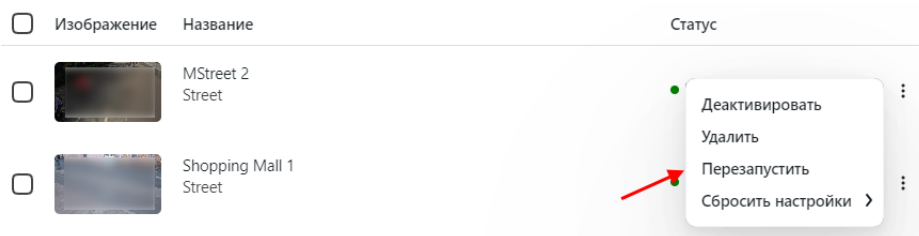
    ...
}
```

Перезапустите контейнер `findface-multi-findface-multi-legacy-1`.

```
sudo docker container restart findface-multi-findface-multi-legacy-1
```

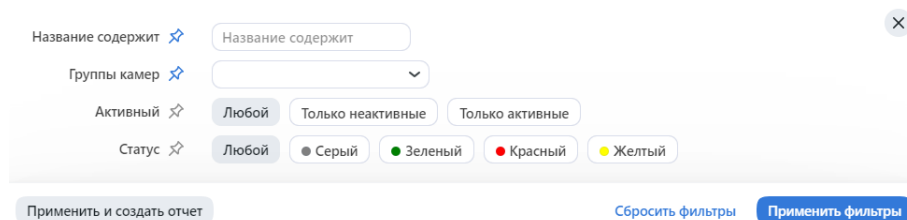
Каждая камера связана с так называемым job-заданием, задачей на обработку видео, содержащей параметры конфигурации и данные видеопотока, которая назначается определенному экземпляру `findface-video-worker`. Данная задача может быть перезапущена.

Для перезапуска job-задания откройте настройки камеры и нажмите на кнопку *Перезапустить*. При этом количество ошибок будет обнулено.



При большом количестве камер в системе используйте следующие фильтры:

- *Название содержит,*
- *Группы камер,*
- *Активный,*
- *Статус.*



3.5 События и эпизоды распознавания объектов

Чтобы отслеживать результат работы системы по части идентификации объектов на видеоизображении в режиме реального времени перейдите на вкладку *Эпизоды и События* и используйте далее вкладки *События* и *Эпизоды*. Помимо работы с текущими событиями идентификации, данные вкладки также предоставляют доступ к истории событий.

Совет: Для идентификации объектов в архивных видео см. *Загрузка и обработка видеофайла*.

3.5.1 Работа с событиями

Данный раздел посвящен вкладке *События*.

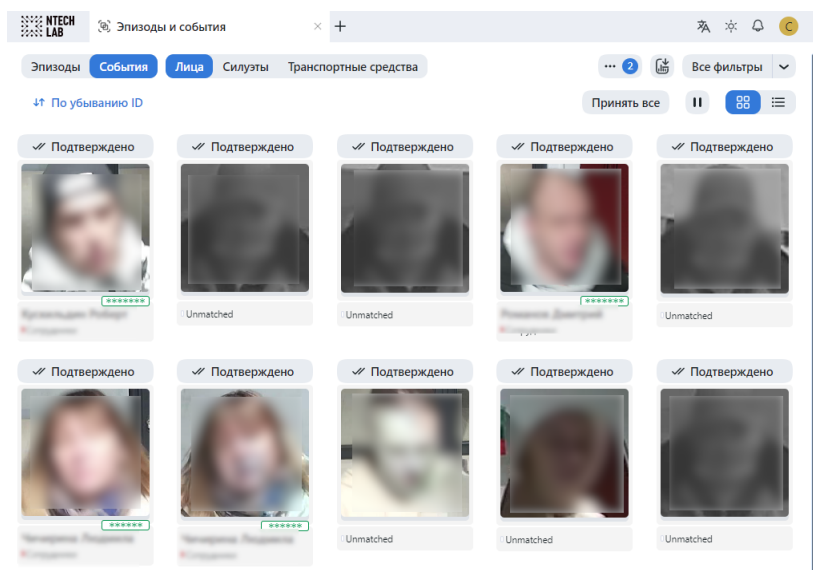
Важно: Вы можете *включить звуковые уведомления* для событий, связанных с конкретными списками наблюдения. В некоторых браузерах для того чтобы воспроизводился звук, вкладка с событиями должна оставаться в фокусе. Чтобы выделить вкладку, откройте ее и щелкните в любом месте страницы.

В этой главе:

- *Просмотр событий*
- *Карточка события. Принятие события*
- *Карточка события. Поиск объекта*
- *Отчеты о рабочем времени по событиям*

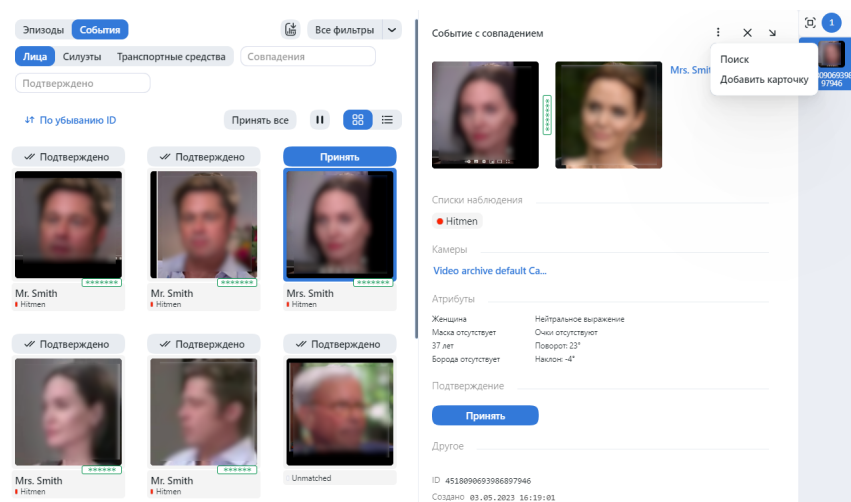
Просмотр событий


Как только объект будет обнаружен, вы увидите уведомление в одном из списков событий: *Лица*, *Силуэты* или *Транспортные средства*, в зависимости от типа объекта.



Уведомление может содержать разную информацию в зависимости от того, есть ли совпадение обнаруженного объекта с карточкой:

- Если совпадение не найдено: нормализованное изображение объекта, дата и время обнаружения объекта, группа камер, списки наблюдения.
- Если совпадение найдено: нормализованное изображение объекта, фотография из карточки, имя, степень схожести объектов, комментарий из карточки, списки наблюдения, дата и время обнаружения объекта, группа камер, атрибуты.



Важно: Для того чтобы остановить вывод новых уведомлений, нажмите на кнопку  над списком событий.

К событиям (уведомлениям) в списке можно применить следующие фильтры:

Примечание: Некоторые фильтры из приведенного ниже списка могут быть скрыты, в зависимости от активированной функциональности распознавания.

- *Объект*: отображать события только для лиц, силуэтов или транспортных средств.
- *Совпадения*: отображать только события с совпадением объектов/без совпадения объектов или все события.
- *Списки наблюдения*: отображать только события по определенному списку наблюдения.
- *Подтверждено*: отображать только подтвержденные/неподтвержденные или все события.
- *Группы камер*: отображать только события по определенной группе камер.
- *Камеры*: отображать только события по определенной камере.
- *Дата и время*: отображать только события, случившиеся в определенный период времени.
- *ID видеоархива*: отобразить события из видеоархива с определенным ID.
- *Имя карточки*: отобразить события с заданным именем карточки.
- *ID эпизода*: отобразить события из эпизода с определенным ID.
- *ID события*: отобразить событие с определенным ID.
- *Лучший кадр события*: отображать все события трека, только события с кадрами в реальном времени, только одно событие с лучшим кадром по окончанию трека.

Специальные фильтры для лиц

- *Возраст*: отобразить события с людьми определенного возраста.
- *Борода*: фильтровать события по наличию бороды.
- *Эмоции*: отобразить события с заданными эмоциями.
- *Пол*: отобразить события с людьми заданного пола.
- *Очки*: фильтровать события по наличию очков на лице.
- *Витальность*: фильтровать события по liveness лица.
- *Медицинская маска*: фильтровать события по наличию медицинской маски.
- *Поворот*: фильтровать события по углу поворота головы.
- *Наклон*: фильтровать события по углу наклона головы.

Специальные фильтры для силуэтов

- *Пол по силуэту*: отобразить события с людьми заданного пола.
- *Возраст по силуэту*: отобразить события с людьми определенного возраста.
- *Головной убор*: отображать только события с людьми в головном уборе заданного типа: шапка/шляпа/кепка, капюшон/платок, без головного убора.
- *Жилет*: отображать только события с людьми, одетыми в жилет заданного цвета.

- *Жилет. Точность распознавания*: отображать только события с людьми, одетыми в жилет с заданной точностью распознавания.
- *Каска*: отображать только события с людьми, одетыми в каску заданного цвета.
- *Каска. Точность распознавания*: отображать только события с людьми, одетыми в каску с заданной точностью распознавания.
- *Цвет верха одежды*: отображать только события с людьми, одетыми в одежду заданного цвета (верх).
- *Цвет низа одежды*: отображать только события с людьми, одетыми в одежду заданного цвета (низ).
- *Тип верха одежды*: отображать только те события, в которых человек носит одежду верха заданного типа: куртка, пальто, жилет, толстовка, футболка, рубашка, платье.
- *Низ одежды*: отображать только те события, в которых человек носит одежду низа заданного типа: брюки, юбка, шорты, неопределенный.
- *Верх одежды*: отображать только те события, в которых человек носит одежду верха заданной обобщенной категории: с длинными рукавами, с короткими рукавами, без рукавов.
- *Сумка на спине*: отображать только события с людьми с сумкой на спине/без сумки на спине.
- *Сумка в руке*: отображать только события с людьми с сумкой в руке/без сумки в руке.

Специальные фильтры для транспортных средств

- *Марка*: фильтровать события по марке транспортного средства.
- *Модель*: фильтровать события по модели транспортного средства.
- *Тип кузова*: отобразить события с транспортными средствами заданного типа кузова.
- *Цвет кузова*: отобразить события с транспортными средствами заданного цвета кузова.
- *Страна*: отобразить события с транспортными средствами, зарегистрированными в заданной стране.
- *Регистрационный номер*: найти транспортное средство с заданным номером.
- *Регион*: отобразить события с транспортными средствами, зарегистрированными в заданном регионе.
- *Цвет номера*: отобразить события с заданным цветом регистрационного номера транспортного средства.
- *Спецтранспорт*: отображать только события с транспортными средствами, принадлежащими заданному типу: полиция, автомобили МЧС и пожарные машины, газоспасательные и аварийно-спасательные службы, скорая помощь, военная техника, коммунальная, строительная техника, прочие.
- *Категория ТС*: отображать только события с транспортными средствами, принадлежащими заданной категории: мотоцикл, скутер, легковой автомобиль с прицепом, грузовик, грузовик с прицепом, автобус, сочлененный автобус и прочие.
- *Вес и размер ТС*: отобразить события с транспортными средствами заданного веса и размера кузова.
- *Ракурс ТС*: отобразить события с транспортными средствами заданного ракурса (спереди, сзади, сбоку).

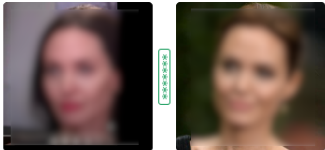
Примечание: Регион и цвет регистрационного знака определяются только для Объединенных Арабских Эмиратов (ОАЭ). Для других стран значения этих атрибутов будут выводиться как неизвестные в результатах распознавания.

Карточка события. Принятие события

Для того чтобы перейти в карточку события из списка событий, щелкните в уведомлении по результату распознавания.

Карточка содержит ту же информацию, что и *уведомление*, а также предоставляет возможность принять событие. Нажмите *Принять*, чтобы поменять статус события на *Подтверждено*.

Событие с совпадением ⋮ × ↗

Mrs. Smith

Списки наблюдения ⋮

● Hitmen

Камеры ⋮

[Video archive default Ca...](#)

Атрибуты ⋮

Женщина	Нейтральное выражение
Маска отсутствует	Очки отсутствуют
37 лет	Поворот: 23°
Борода отсутствует	Наклон: -4°

Подтверждение ⋮

Принять

Другое ⋮

ID 4518048241513281717

Создано 03.05.2023 11:55:27

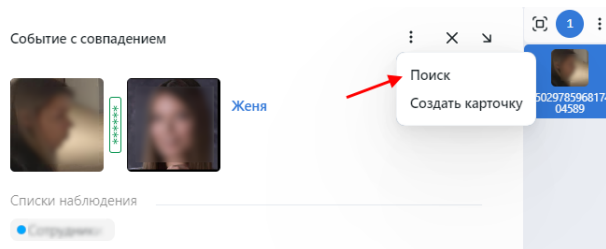
Совет: Если у обнаруженного объекта есть совпадения по картотеке, в соответствующую карточку можно перейти, щелкнув по имени в карточке события.

Совет: Для того чтобы принять все события, нажмите на кнопку *Принять все* над списком событий.

Примечание: Принятие события может быть автоматизировано для выбранных списков наблюдения.

Карточка события. Поиск объекта

FindFace Multi позволяет искать обнаруженные объекты в списке событий и в картотеке. Для перехода на вкладку поиска из карточки события нажмите *Поиск*.



См.также:

- *Поиск объектов в системе*

Отчеты о рабочем времени по событиям

См. *Отчеты*.

3.5.2 Эпизоды событий

Данный раздел посвящен вкладке *Эпизоды*.

См.также:

- *Работа с событиями*

Эпизод — это набор событий идентификации, в которых фигурируют объекты одного происхождения (изображения лица и силуэта одного и того же человека и изображения одного и того же транспортного средства), обнаруженные в течение определенного периода времени. Поскольку события на вкладке *События* отображаются в произвольном порядке, обработка большого количества разнородных событий может быть делом затруднительным и неэффективным. Функция эпизодов позволяет автоматически объединять входящие события на основе времени обнаружения и схожести объектов. Это позволяет с легкостью обрабатывать разнородные события даже в больших количествах.

В этой главе:

- *Об эпизодах*
- *Назначение прав на эпизоды*
- *Просмотр эпизодов*
- *Принятие события и эпизода*
- *Параметры эпизода*

Об эпизодах

В жизненном цикле эпизода существует два этапа:

- LIVE: открытый на данный момент эпизод, в который могут добавлены новые события.
- Закрытый: закрытый эпизод, добавление событий невозможно.

Эпизоды классифицируются как эпизоды с людьми и эпизоды с транспортными средствами. Эпизоды с людьми содержат изображения лиц и силуэтов людей, обнаруженные в течение определенного периода времени, а эпизоды с транспортными средствами содержат изображения транспортного средства и регистрационного номера.

Назначение прав на эпизоды

Пользователь получает уведомление о новом эпизоде, если у него есть права на открывающее этот эпизод событие. Просмотр новых событий в эпизоде также требует соответствующих прав.

Право на событие состоит из прав на соответствующие камеру и список наблюдения.

Примечание: Для просмотра не сопоставленных с карточкой событий вам понадобятся только права на камеру.

Для управления правами на объект **Эпизод** перейдите в разрешения для соответствующей роли и настройте следующие разрешения:

- `humanepisode`: эпизоды с людьми
- `carepisode`: эпизоды с транспортными средствами

Совет: См. *Управление ролями и пользователями*.

User ⋮ × ↗

Сведения Списки наблюдения Группы камер Права доступа

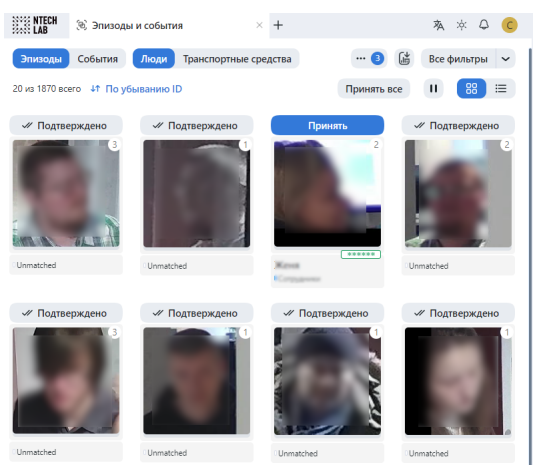
121 всего, 80 выкл.

Название	Прос...	Изме...	Доба...	Удалить
all_own_sessions	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
area	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
bodycluster	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
bodyevent	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
bodyobject	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
camera	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
cameragroup	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
carcard	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
carcluster	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
carepisode	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
carevent	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
carobject	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
case	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
counter	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
deviceblacklistrecord	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
facecluster	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Просмотр эпизодов

Вы можете найти список эпизодов с фильтрами и статистикой на вкладке *Эпизоды*. Эпизоды сортируются по происхождению объекта: есть эпизоды с людьми и эпизоды с транспортными средствами.

Событие обнаружения объекта либо добавляется в существующий LIVE-эпизод, либо инициирует создание нового эпизода. Каждому эпизоду присваивается id, который впоследствии можно использовать для фильтрации событий и эпизодов.



К эпизодам в списке можно применить следующие фильтры:

Примечание: Некоторые фильтры из приведенного ниже списка могут быть скрыты, в зависимости

от активированной функциональности распознавания.

- *Объект*: отобразить только эпизоды с людьми или только эпизоды с транспортными средствами.
- *Совпадения: лица*: отобразить только эпизоды с совпадением лиц/без совпадения лиц/все эпизоды.
- *Совпадения: силуэты*: отобразить только эпизоды с совпадением силуэтов/без совпадения силуэтов/все эпизоды.
- *Совпадения*: отобразить только эпизоды с совпадением транспортных средств/без совпадения транспортных средств/все эпизоды.
- *Списки наблюдения*: отобразить только эпизоды по определенному списку наблюдения.
- *Подтверждено*: отобразить только подтвержденные/неподтвержденные или все эпизоды.
- *Группы камер*: отобразить только эпизоды по определенной группе камер.
- *Камеры*: отобразить только эпизоды по определенной камере.
- *Дата и время*: отобразить только эпизоды, случившиеся в определенный период времени.
- *ID видеоархива*: отобразить эпизоды из видеоархива с определенным ID.
- *Имя карточки*: отобразить эпизоды с заданным именем карточки.
- *ID эпизода*: отобразить эпизод с определенным ID.
- *Количество событий*: отобразить эпизоды с заданным количеством событий.

Специальные фильтры для эпизодов с лицами

- *Возраст*: отобразить эпизоды с людьми определенного возраста.
- *Борода*: фильтровать эпизоды по наличию бороды.
- *Эмоции*: отобразить эпизоды с заданными эмоциями.
- *Пол*: отобразить эпизоды с людьми заданного пола.
- *Очки*: фильтровать эпизоды по наличию очков на лице.
- *Витальность*: фильтровать эпизоды по liveness лица (истинное лицо или изображение).
- *Медицинская маска*: фильтровать эпизоды по наличию медицинской маски.

Специальные фильтры для эпизодов с силуэтами

- *Пол по силуэту*: отобразить эпизоды с людьми заданного пола.
- *Возраст по силуэту*: отобразить эпизоды с людьми определенного возраста.
- *Головной убор*: отобразить только эпизоды с людьми в головном уборе заданного типа: шапка/шляпа/кепка, капюшон/платок, без головного убора.
- *Жилет*: отобразить только эпизоды с людьми, одетыми в жилет заданного цвета.
- *Жилет. Точность распознавания*: отобразить только эпизоды с людьми, одетыми в жилет с заданной точностью.
- *Каска*: отобразить только эпизоды с людьми в каске заданного цвета.

- *Каска. Точность распознавания*: отобразить только эпизоды с людьми в каске с заданной точностью распознавания.
- *Цвет верха одежды*: отобразить только эпизоды с людьми, одетыми в одежду заданного цвета (верх).
- *Цвет низа одежды*: отобразить только эпизоды с людьми, одетыми в одежду заданного цвета (низ).
- *Тип верха одежды*: отобразить только те эпизоды, в которых человек носит одежду верха заданного типа: куртка, пальто, безрукавка, толстовка, футболка, рубашка, платье.
- *Низ одежды*: отобразить только те эпизоды, в которых человек носит одежду низа заданного типа: брюки, юбка, шорты, неопределенный.
- *Верх одежды*: отобразить только те эпизоды, в которых человек носит одежду верха заданной обобщенной категории: с длинными рукавами, с короткими рукавами, без рукавов.
- *Сумка на спине*: отображать только эпизоды с людьми с сумкой на спине/без сумки на спине.
- *Сумка в руке*: отображать только эпизоды с людьми с сумкой в руке/без сумки в руке.

Специальные фильтры для эпизодов с транспортными средствами


- *Марка*: фильтровать эпизоды по марке транспортного средства.
- *Модель*: фильтровать эпизоды по модели транспортного средства.
- *Тип кузова*: отобразить эпизоды с транспортными средствами заданного типа кузова (минивэн, лимузин и др.).
- *Цвет кузова*: отобразить эпизоды с транспортными средствами заданного цвета кузова.
- *Страна*: отобразить эпизоды с транспортными средствами, зарегистрированными в заданной стране.
- *Регистрационный номер*: найти эпизод с заданным номером транспортного средства.
- *Регион*: отобразить эпизоды с автомобилями, зарегистрированными в заданном регионе.
- *Цвет номера*: отобразить эпизоды с заданным цветом регистрационного номера транспортного средства.
- *Спецтранспорт*: отображать только эпизоды с транспортными средствами, принадлежащими заданному типу: такси, маршрутный транспорт, каршеринг, скорая помощь, полиция, автомобили МЧС и пожарные машины, газоспасательные и аварийно-спасательные службы, военная техника, коммунальная, строительная техника, прочие.
- *Категория ТС*: отображать только эпизоды с транспортными средствами, принадлежащими заданной категории: мотоцикл, скутер, легковой автомобиль с прицепом, грузовик, грузовик с прицепом, автобус, сочлененный автобус, прочие.
- *Вес и размер ТС*: отобразить эпизоды с транспортными средствами заданного веса и размера кузова.

Примечание: Регион и цвет регистрационного знака определяются только для Объединенных Арабских Эмиратов (ОАЭ). Для других стран значения этих атрибутов будут выводиться как неизвестные в результатах распознавания.

Для просмотра событий эпизода щелкните по нужному эпизоду в списке. Вы будете переправлены на вкладку *Эпизод с совпадением* или *Эпизод без совпадения* со сведениями об эпизоде и связанными событиями.

Эпизод с совпадением ✕ ↗

Сведения События



Списки наблюдения

Камеры

Атрибуты

33 года	Отсутствует
Нейтральное выражение	Женский
Очки отсутствуют	-37
-55	Маска отсутствует

Подтверждение

Принять

Другое

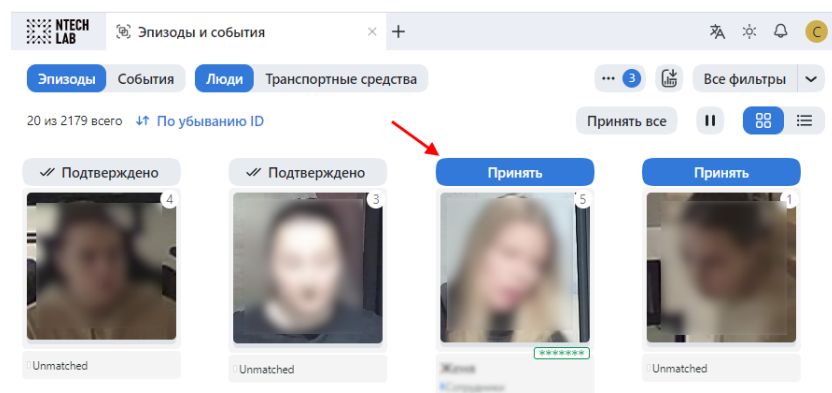
ID 1898

Создано 27.02.2023 16:03:04

Работа с вкладкой *События* описана в разделе *Работа с событиями*.

Принятие события и эпизода

Для того чтобы подтвердить эпизод целиком, нажмите *Принять* в списке эпизодов. В результате все события в эпизоде будут автоматически подтверждены, включая события, которые еще не добавлены (в случае эпизода LIVE).



NTECH LAB Эпизоды и события

Эпизоды События Люди Транспортные средства

20 из 2179 всего По убыванию ID

Принять все

Подтверждено 4

Подтверждено 3

Принять 5

Unmatched

Unmatched

Женский

Unmatched 1

Эпизод также автоматически подтверждается, если вы приняли все события по отдельности.

Параметры эпизода

Для настройки эпизодов вам понадобится файл конфигурации `/opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py`. Найдите следующие параметры в секции `FFSECURITY`:

- `FACE_EPISODES_THRESHOLD`, `BODY_EPISODES_THRESHOLD`, `CAR_EPISODES_THRESHOLD`: пороговые значения схожести, которые система использует при поиске недавних событий для формирования эпизода. По умолчанию установлены оптимальные значения. При необходимости вы можете изменить их. Обязательно предварительно проконсультируйтесь с нашими техническими специалистами (support@ntechlab.com).
- `FACE_EPISODE_SEARCH_INTERVAL`, `BODY_EPISODE_SEARCH_INTERVAL`, `CAR_EPISODE_SEARCH_INTERVAL`: период, предшествующий событию, за который система ищет уже существующие события с похожими объектами в базе данных вектора признаков. Если такое событие не найдено, система создает новый эпизод. В противном случае она сортирует 100 самых последних похожих объектов и выбирает наиболее подходящее событие из эпизода `LIVE`.
- `EPISODE_MAX_DURATION`: максимальная продолжительность эпизода в секундах. По истечении этого времени эпизод автоматически закрывается.
- `EPISODE_EVENT_TIMEOUT`: максимальное время в секундах с момента добавления последнего события в эпизод. По истечении этого времени эпизод автоматически закрывается.
- `EPISODE_KEEP_ONLY_BEST_EVENT`: при закрытии эпизода, удалять в нем все события, за исключением события с лучшим объектом. Использование данной опции помогает экономить дисковое пространство.

```
sudo vi /opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py

...

FFSECURITY = {
    ...

    # episodes specific matching threshold that is used to join faces in an episode
    'FACE_EPISODES_THRESHOLD': 0.673, # FAR = 1.00E-06 # model: [mango_320]
    'BODY_EPISODES_THRESHOLD': 0.75, # model: [clio]
    'CAR_EPISODES_THRESHOLD': 0.61, # model: [alonso]
    ...
    # when closing episode, delete all events except the best episode event
    'EPISODE_KEEP_ONLY_BEST_EVENT': False,
    # delete episode events after delay in seconds
    'EPISODE_DELETE_EVENTS_DELAY': 60,
    ...
    # maximum event age in seconds than could be added to an episode.
    'FACE_EPISODE_SEARCH_INTERVAL': 60,
    'BODY_EPISODE_SEARCH_INTERVAL': 60,
    'CAR_EPISODE_SEARCH_INTERVAL': 60,
    # maximum episode duration (episode is closed after)
    'EPISODE_MAX_DURATION': 300,
    # if no new event added to an episode during this timeout, episode will be closed.
    'EPISODE_EVENT_TIMEOUT': 30,
    ...
}
```

(continues on next page)

(продолжение с предыдущей страницы)

...

Перезапустите контейнер `findface-multi-findface-multi-legacy-1`.

```
sudo docker container restart findface-multi-findface-multi-legacy-1
```

3.6 Поиск объектов в системе

FindFace Multi позволяет выполнять поиск объектов по всей системе.

Для того чтобы найти человека, выполните следующие действия:

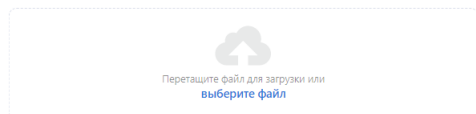
1. Перейдите на вкладку *Поиск*.
2. Задайте искомый объект одним из следующих способов:
 - задав URL или ID события
 - задав URL или ID карточки
 - задав URL или ID кластера
 - загрузив фотографию

Задайте источник поиска

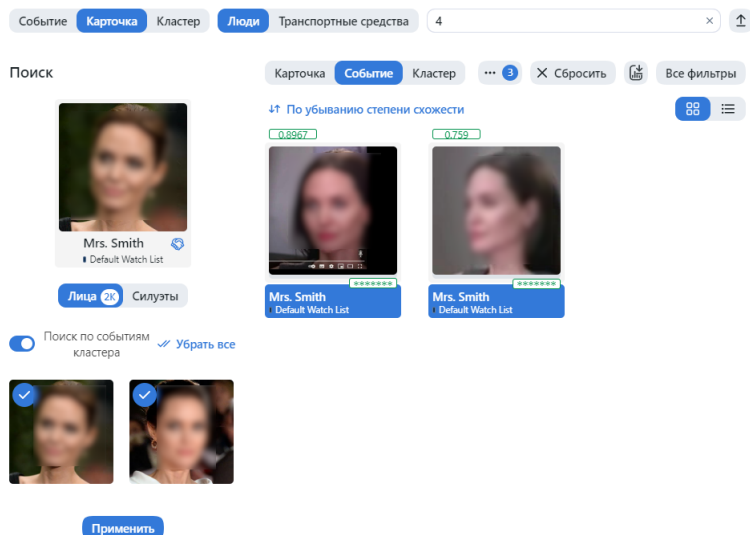
Введите URL фото в Интернете или ID/URL внутренней сущности

Событие Карточка Кластер Введите ID или URL

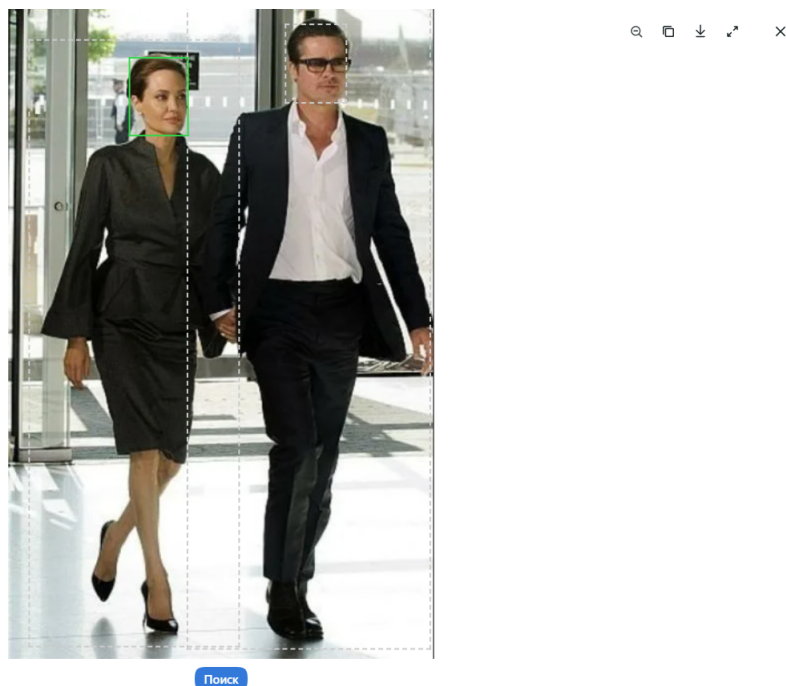
или загрузите медиафайл. Поддерживаемые форматы: JPG, PNG, GIF



3. В случае если вы задали URL карточки, выберите из нее фотографию. Если в карточке несколько фотографий, вы можете выбрать некоторые или все из них. Нажмите на кнопку *Применить*.



4. В случае если была загружена фотография, вы увидите ее в новом окне. Если на фотографии присутствует несколько объектов, выберите нужное. Нажмите на кнопку *Поиск*.

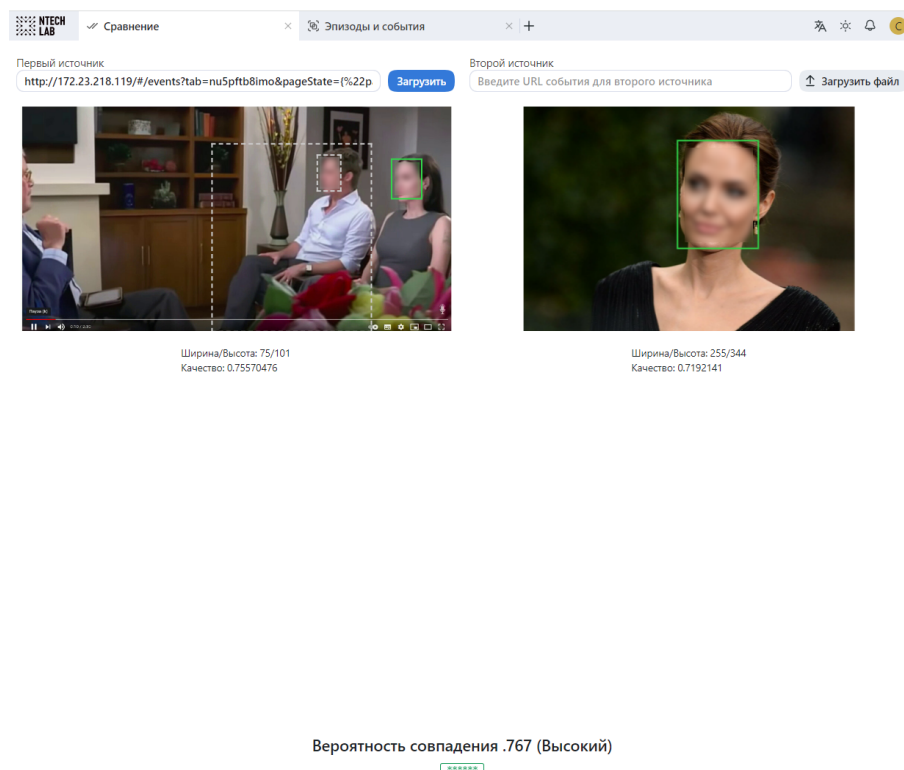


5. Вы увидите результаты поиска. При необходимости можно сузить область поиска, указав список наблюдения, порог сходства и т. д.

3.7 Сравнение двух объектов

FindFace Multi позволяет сравнить два объекта и убедиться, что они совпадают. Выполните следующие действия:

1. Перейдите на вкладку *Сравнение*.
2. Укажите два объекта для проверки одним из следующих способов:
 - задав URL события
 - загрузив фотографию
3. Если на изображении несколько объектов, выберите тот, который вас интересует.



4. В результате будет отображена степень схожести между ними.

3.8 Счетчики лиц, силуэтов и транспортных средств. Определение дистанции

Важно: Для подсчета человеческих силуэтов или автомобилей нужно предварительно включить детектирование *силуэтов* или *транспортных средств*.

FindFace Multi позволяет подсчитывать лица, силуэты и транспортные средства на подключенных камерах. Данную функциональность можно использовать в различных целях, например, для подсчета людей в очередях и зонах ожидания, мониторинга общественных собраний, предотвращения скопления людей, определение пробок на дорогах и т. д.

Метод подсчета основан на временных срезах. Это означает, что счетчик считает лица, силуэты и транспортные средства на статических скриншотах, которые делаются с заданным интервалом.

Вы можете вести подсчет лиц/силуэтов/транспортных средств на каждой отдельной камере или же совместно на всех выбранных камерах.

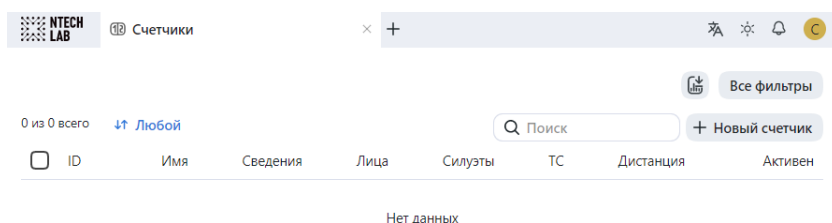
В этом разделе:

- *Создание счетчика*
- *Калибровка счетчика для определения дистанции*
- *Мониторинг работы счетчиков*
- *Работа с записями счетчика*
- *Настройка вебхука для счетчика*
- *Настройка счетчиков*

3.8.1 Создание счетчика

Для создания счетчика выполните следующие действия:

1. Перейдите на вкладку *Счетчики*.
2. Нажмите *+ Новый счетчик*.



3. Укажите имя счетчика.
4. Выберите одну или несколько групп камер для подсчета.
5. Выберите одну или несколько камер для подсчета.
6. Установите флажок *Лица* для подсчета лиц.
7. Установите флажок *Силуэты* для подсчета силуэтов. Детектирование силуэтов должно быть *включено*.
8. Установите флажок *Транспортные средства* для подсчета транспортных средств. Детектирование транспортных средств должно быть *включено*.
9. Задайте интервал в секундах между двумя последовательными скриншотами, используемыми для подсчета лиц/силуэтов/транспортных средств..
10. Нажмите *Сохранить*.
11. Установите флажок *Определение дистанции* для измерения расстояния между силуэтами. Данный флажок доступен в том случае, если установлен флажок *Силуэты*.

Enter_2
Enter_2

Скриншоты **Сведения** ROI

Название
Enter_2

Группы камер
Office X

Камеры
qa2 X

Детекторы

☒ Лица ☒ Силуэты ☐ Транспортные средства

Интервал счета, секунды
1

☒ Определение дистанции

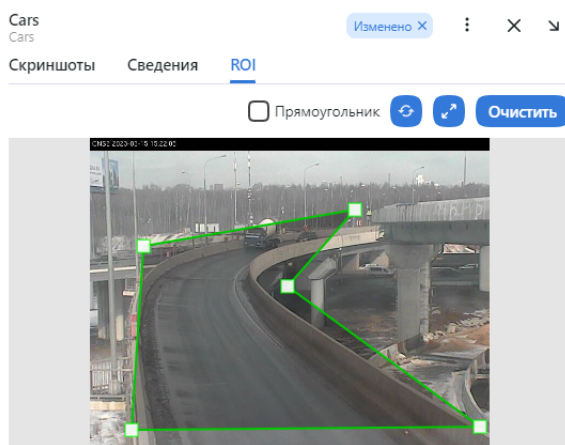
Статус

Не калиброван

60 **Откалибровать счетчик** Удалить калибровку

ID 6
Создано 15.03.2023 15:25:44
Обновлено 15.03.2023 15:25:51

12. Убедитесь, что для счетчика установлен флажок *Активен*.
13. Нажмите *Сохранить*.
14. (Опционально) Перейдите на вкладку *ROI*, чтобы задать регион отслеживания лиц/силуэтов/транспортных средств в поле зрения камер(ы). Нажмите *Сохранить*.



3.8.2 Калибровка счетчика для определения дистанции

Если включено *Определение дистанции*, необходимо выполнить калибровку счетчика.

Для калибровки счетчика:

1. Попросите человека встать перед выбранной камерой в полный рост.
2. Попросите пройти по зоне, фиксируемой камерой, по которой в дальнейшем будут рассчитываться расстояния. Для достижения наилучшего качества человек должен ходить по ровной поверхности.
3. Введите период калибровки счетчика (от 15 до 300 секунд) и нажмите *Откалибровать счетчик*.

Если нужно откалибровать счетчик для нескольких камер, выполните шаги 1-3 для каждой камеры, выбранной в поле *Камеры*.

Если результат калибровки вас не удовлетворил, нажмите *Удалить калибровку* и выполните калибровку еще раз.

3.8.3 Мониторинг работы счетчиков

Мониторинг работы счетчиков выполняется на вкладке *Счетчики*.

Статусы счетчиков:

- Зеленый: счетчик работает без ошибок, или количество ошибок не превышает допустимого порогового значения.
- Желтый: количество ошибок превышает пороговое значение.
- Красный: количество ошибок является критическим.
- Серый: счетчик отключен.

Совет: Вы можете настроить желтый и красный статусы на основании доли неудачных записей, а также изменить продолжительность временного окна между двумя последовательными проверками состояния счетчика. Для этого отредактируйте следующие параметры в файле конфигурации `/opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py`:

```
sudo vi /opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py

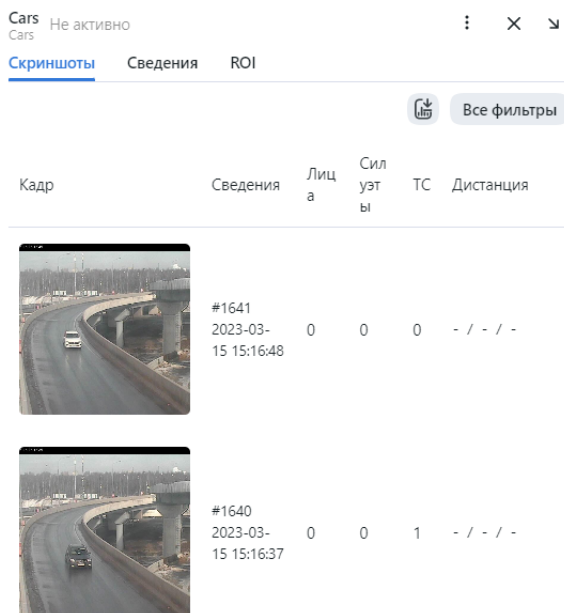
FFSECURITY = {
    ...
    # Counter health status config:
    # max percent of camera records with errors
    'MAX_COUNTER_ERROR_RECORDS': {'yellow': 0.3, 'red': 0.5},
    # time window for computing health status (in seconds)
    'COUNTER_HEALTH_STATUS_TIME_WINDOW': 30,
    ...
}
```

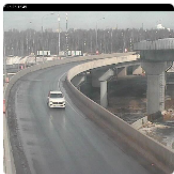

3.8.4 Работа с записями счетчика

Статические скриншоты, которые делаются счетчиком, с количеством обнаруженных лиц, силуэтов и транспортных средств, сохраняются как т.н. записи счетчика. Если вы включили для счетчика измерение расстояний, каждая запись также будет содержать минимальное, среднее и максимальное обнаруженные расстояния в метрах.

Если счетчик работает с ошибками, система будет создавать пустые записи с сообщением об ошибке.

Для просмотра записей счетчика перейдите на вкладку *Счетчики*. Нажмите на счетчик. Перейдите на вкладку *Скриншоты*.



Кадр	Сведения	Лица	Силуэты	ТС	Дистанция
	#1641 2023-03-15 15:16:48	0	0	0	- / - / -
	#1640 2023-03-15 15:16:37	0	0	1	- / - / -

Вы можете увеличить скриншот, нажав на него. Если включено определение дистанции, на скриншоте будут зафиксированы прямоугольники вокруг силуэтов и расстояние между ними.

Для работы со списком счетчиков используйте следующие фильтры:

- Название содержит
- ID счетчика
- Камеры
- Группы камер
- Дистанция

Для работы со скриншотами счетчика используйте следующие фильтры:

- Камеры
- Дата и время
- Считать лица
- Считать силуэты
- Считать транспортные средства
- ID записи счетчика
- Минимальная дистанция

- Максимальная дистанция
- Средняя дистанция

3.8.5 Настройка вебхука для счетчика

Для того чтобы автоматизировать процесс сбора статистики по счетчику, *настройте вебхук* на срабатывание при определенном количестве лиц, силуэтов и транспортных средств в записи счетчика.

См.также:

- *Распознавание силуэтов и их атрибутов*
- *Распознавание транспортных средств и их атрибутов*
- *Вебхуки*

3.8.6 Настройка счетчиков

Для настройки счетчиков откройте файл конфигурации `/opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py` и измените следующие параметры:

- **COUNTERS_DEDUP_OPTIONS**: набор опций, которые позволяют избежать дублирования объектов при подсчете в многокамерных счетчиках, т. е. ситуации, когда один и тот же объект считается на нескольких камерах одновременно, что ведет к чрезмерно завышенному результату подсчета. Для каждого типа объектов доступны две опции (**face**, **body**, **car**): **enabled** - включает дедупликацию объектов, **threshold** - определяет порог схожести, при котором объекты считаются дубликатами.
- **COUNTERS_SAVE_FULLFRAME** определяет параметры сохранения полных кадров при работе счетчиков: **always** (всегда), **detect** - сохранять, только если были обнаружены лица, силуэты или транспортные средства, **never** (никогда).
- **COUNTERS_FULLFRAME_JPEG_QUALITY**: качество JPEG полных кадров.
- **COUNTERS_THUMBNAIL_JPEG_QUALITY**: качество JPEG миниатюр.
- **COUNTERS_ROI_INTERSECTION_THRESHOLD**: требуемый процент пересечения между прямоугольником вокруг объекта (**bbox**) и областью интереса, определенной для счетчика.

```
sudo vi /opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py

# deduplicate objects on multi-camera counters
'COUNTERS_DEDUP_OPTIONS': {
    'face': {
        'enabled': False,
        'threshold': 0.714, # model: [mango_320]
    },
    'body': {
        'enabled': False,
        'threshold': 0.65, # model: [clio]
    },
    'car': {
        'enabled': False,
        'threshold': 0.65, # model: [alonso]
    },
},
```

(continues on next page)

(продолжение с предыдущей страницы)

```
# counters full frame saving options:
# `always` - save always
# `detect` - save only if faces or bodies have been detected
# `never` - never save full frames
'COUNTERS_SAVE_FULLFRAME': 'always',
'COUNTERS_FULLFRAME_JPEG_QUALITY': 75,
'COUNTERS_THUMBNAIL_JPEG_QUALITY': 75,
# required percentage of bbox intersection with ROI
'COUNTERS_ROI_INTERSECTION_THRESHOLD': 0.75,
...
```

После внесения изменений обязательно перезапустите контейнер `findface-multi-findface-multi-legacy-1`.

```
sudo docker container restart findface-multi-findface-multi-legacy-1
```

3.9 Кластеры лиц, силуэтов, транспортных средств

FindFace Multi поддерживает автоматическую кластеризацию объектов одного происхождения:

- Изображения лиц, принадлежащие одному и тому же человеку, образуют кластер лиц.
- Изображения силуэтов, принадлежащие одному и тому же человеку, образуют кластер силуэтов.
- Изображения одного и того же транспортного средства образуют кластер транспортных средств.

Совокупные галереи кластеров лиц, силуэтов и транспортных средств доступны на вкладке *Кластеры*.

Примечание: Если для кластера лиц/силуэтов найдено совпадение с *карточкой человека*, он автоматически отобразится в данной карточке. По аналогии, кластер транспортных средств будет сохранен в соответствующей карточке транспортного средства.

Важно: По умолчанию кластеризация объектов отключена. *Включите и настройте ее* через файл конфигурации `/opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py`.

В этом разделе:

- *Алгоритмы кластеризации*
- *Включение и настройка кластеризации*
- *Работа с галереями кластеров*
 - *Просмотр кластеров. Фильтры*
 - *Объединение и удаление кластеров*
- *Статическая галерея кластеров*
- *Ручная кластеризация*

3.9.1 Алгоритмы кластеризации

FindFace Multi использует следующие алгоритмы кластеризации объектов:

- Кластеризация в реальном времени. Данный алгоритм кластеризации обрабатывает эпизоды с целью выбора подходящих изображений объектов и кластеризует выбранные изображения. Он работает на лету после закрытия эпизода. Результаты кластеризации отображаются динамически на вкладке *Кластеры* и в соответствующей *карточке*.

Для кластеризации используются не все эпизоды. Если эпизод соответствует всем требованиям (см. подробности *ниже*), система формирует кластер следующим образом:

- Выбирает событие наилучшего качества.
- Создает новую сущность `cluster event` (кластерное событие) в основной базе данных PostgreSQL. Сущность содержит метаданные выбранного события, вектор признаков объекта и миниатюру объекта, а также ссылку на родительский эпизод.
- Ищет похожий центроид объекта в галерее `cluster_events` базы данных векторов признаков Tarantool. Центроид объекта — это виртуальный вектор признаков, усредненный по всем объектам-близнецам, которые были обнаружены на данный момент (например, центроид лица — это вектор признаков, усредненный по всем изображениям лица одного и того же человека). Система обновляет похожий центроид, используя новое событие, если такой центроид найден. Иначе создается новый центроид.
- Кластеризация по расписанию. Данный алгоритм кластеризации перерабатывает и пересматривает кластерные события, созданные во время кластеризации в реальном времени. Использование данного алгоритма улучшает качество кластерного центроида, поскольку в этом случае центроид усредняется по большему массиву накопленных векторов признаков. Результаты кластеризации по расписанию отображаются после каждой запланированной итерации на вкладке *Кластеры* и в соответствующей *карточке*.

Расписание задается в формате RRULE в параметре `CLUSTERS_CLUSTERIZATION_SCHEDULE` файла конфигурации `/opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py`. Предпочтительны ночные часы, так как запланированная кластеризация занимает много времени и ресурсов ЦП.

Важно: Кластеризация по расписанию полностью перезаписывает содержимое галерей кластеров, включая идентификаторы. Вы можете закрепить определенные кластеры, включив настройки `CLUSTERS_AUTO_PIN_HEURISTICS` и `PIN_MATCHED_CLUSTERS` (см. *ниже*).

3.9.2 Включение и настройка кластеризации

По умолчанию кластеризация отключена. Чтобы включить и настроить ее, выполните следующие действия:

1. Откройте файл конфигурации `/opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py`. Найдите раздел `Clusters configuration`.

```
sudo vi /opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py

# -- Clusters configuration --
'ENABLE_NIGHT_CLUSTERIZATION': False,
'ENABLE_REALTIME_CLUSTERIZATION': True,
# rrule (recurrence rule) for scheduling "night" clusterization
```

(continues on next page)

(продолжение с предыдущей страницы)

```

'CLUSTERIZATION_SCHEDULE': 'RRULE:FREQ=DAILY;INTERVAL=1;WKST=MO;BYHOUR=0;BYMINUTE=0
→',
# clusterize only selected objects types (for realtime and nightly clusterization)
# available are: face, body, car
'CLUSTERIZE_OBJECT_TYPES': ['face'],
# keep no more than N the best quality events in centroid (None to disable)
'CENTROID_MAX_SIZE': None,
# save cluster events without emben and images (only keep thumbnail for the best
→event)
'LONG_LIVING_CLUSTER_EVENTS': False,
# skip clusterization if unpinned cluster events count is greater than this value
'CLUSTERIZATION_MAX_CLUSTER_EVENTS': None,
# create cluster only from cluster events in current case or only from cluster
→events without case
'ISOLATE_CASE_CLUSTERS': True,
# cluster event to cluster matching confidence threshold
'FACE_CLUSTER_CONFIDENCE_THRESHOLD': 0.714, # model: [mango_320]
'BODY_CLUSTER_CONFIDENCE_THRESHOLD': 0.65, # model: [clio]
# minimum required event quality for cluster creation
'FACE_CLUSTER_EVENT_MIN_QUALITY': 0.5, # model: [quality_fast.v1]
'BODY_CLUSTER_EVENT_MIN_QUALITY': 0.6, # model: [pedattr.quality.v0]
'CAR_CLUSTER_EVENT_MIN_QUALITY': 0.73, # model: [carattr.quality.v0]
# discard cluster event if `max_centroids` similar centroids found with confidence
→greater than `confidence`
'FACE_CLUSTER_MAX_N_SIMILAR': {'enabled': False, 'max_centroids': 5, 'confidence':
→0.714},
'BODY_CLUSTER_MAX_N_SIMILAR': {'enabled': False, 'max_centroids': 5, 'confidence':
→0.65},
# minimum required object size in pixels for cluster creation
'FACE_CLUSTER_EVENT_MIN_SIZE': 50,
'BODY_CLUSTER_EVENT_MIN_SIZE': 50,
'CAR_CLUSTER_EVENT_MIN_SIZE': 50,
# minimum required number events in episode for cluster creation
'FACE_CLUSTER_EVENT_MIN_EPISODE_EVENTS': 1,
'BODY_CLUSTER_EVENT_MIN_EPISODE_EVENTS': 1,
'CAR_CLUSTER_EVENT_MIN_EPISODE_EVENTS': 1,
# age feature threshold for cluster creation
'FACE_CLUSTER_EVENT_MIN_AGE_THRESHOLD': 16,
....
# pinned clusters keep their id and events after reclusterization
'CLUSTERS_AUTO_PIN_HEURISTICS': {
  'face': {
    # pin clusters with `value` minimum cluster events
    'min_events': {'enabled': True, 'value': 10},
    # cluster's centroid similarity confidence is less than
    'max_centroid_similarity_threshold': {'enabled': True, 'value': 0.54}, #
→mango_320
    # minimum average event's quality
    'min_average_events_quality': {'enabled': True, 'value': 0.45},
  },
  'body': {},
  'car': {},

```

(continues on next page)

(продолжение с предыдущей страницы)

```

    },
    # always pin clusters with matched events (not affected by heuristics above)
    'PIN_MATCHED_CLUSTERS': False,
    ...

```

2. Включите кластеризацию в реальном времени, установив `ENABLE_REALTIME_CLUSTERIZATION: True`.
3. При необходимости включите кластеризацию по расписанию, установив `ENABLE_NIGHT_CLUSTERIZATION: True`.

Важно: Включать кластеризацию по расписанию имеет смысл только в том случае, если включена кластеризация в реальном времени. В противном случае система не сформирует ни одного нового кластера, поскольку только кластеризация в реальном времени является поставщиком **уникальных** кластерных событий.

```

...
# -- Clusters configuration --
'ENABLE_NIGHT_CLUSTERIZATION': True,
'ENABLE_REALTIME_CLUSTERIZATION': True,
...

```

4. При необходимости задайте рекуррентное правило (RRULE), задающее расписание плановой кластеризации. Если правило не задано, кластеризация автоматически начинается в 00:00 GMT.

Совет: См. калькулятор RRULE.

```

# rrule (recurrence rule) for scheduling clusters clusterization
'CLUSTERIZATION_SCHEDULE': 'RRULE:FREQ=DAILY;INTERVAL=1;WKST=MO;BYHOUR=0;BYMINUTE=0
→',

```

5. По умолчанию система формирует только кластеры лиц. Для того чтобы включить формирование кластеров силуэтов и транспортных средств, добавьте соответствующие типы объектов в следующую строку:

```

# available are: face, body, car
'CLUSTERIZE_OBJECT_TYPES': ['face', 'body', 'car'],

```

6. При необходимости измените минимальное количество событий в используемых для кластеризации эпизодах. По умолчанию данное количество равно 1. Сделайте это отдельно для каждого типа объекта.

```

# minimum required number events in episode for cluster creation
'FACE_CLUSTER_EVENT_MIN_EPISODE_EVENTS': 3,
'BODY_CLUSTER_EVENT_MIN_EPISODE_EVENTS': 3,
'CAR_CLUSTER_EVENT_MIN_EPISODE_EVENTS': 2,

```

7. При необходимости измените минимальное качество используемых для кластеризации изображений объектов. Сделайте это отдельно для каждого типа объекта.

Примечание: Поскольку данная настройка требует наличия высокого уровня знаний и опыта,

мы настоятельно рекомендуем предварительно проконсультироваться с нашими техническими специалистами.

```
# minimum required event quality for cluster creation
'FACE_CLUSTER_EVENT_MIN_QUALITY': 0.5, # model: [quality_fast.v1]
'BODY_CLUSTER_EVENT_MIN_QUALITY': 0.6, # model: [pedattr.quality.v0]
'CAR_CLUSTER_EVENT_MIN_QUALITY': 0.73, # model: [carattr.quality.v0]
```

8. При необходимости измените порог уверенности алгоритма в совпадении объектов при сопоставлении кластерного события и кластера.

Предупреждение: Перед изменением данного параметра проконсультируйтесь с нашими специалистами по адресу support@ntechlab.com.

```
# cluster event to cluster matching confidence threshold
'FACE_CLUSTER_CONFIDENCE_THRESHOLD': 0.714, # model: [mango_320]
'BODY_CLUSTER_CONFIDENCE_THRESHOLD': 0.65, # model: [clio]
```

9. Кластеризация по расписанию полностью перезаписывает все созданные кластеры. Вы можете «закрепить» определенные кластеры, т. е. сохранить их и связанные кластерные события, включая идентификаторы, нетронутыми. Для этого используйте следующие настройки:

Примечание: Данные настройки являются независимыми. При необходимости используйте обе.

Примечание: Данные настройки не влияют на кластеризацию в реальном времени. Она продолжит создавать новые кластерные события для закрепленных кластеров.

- **CLUSTERS_AUTO_PIN_HEURISTICS:** установите **True** или **False** для следующих параметров и укажите соответствующие значения:

Примечание: При необходимости сделайте это для каждого типа объекта.

- **min_events:** закрепить кластер, когда количество связанных с ним кластерных событий превысит заданное минимальное значение.
- **max_centroid_similarity_threshold:** закрепить кластер, если сходство между его центроидом и центроидами других кластеров меньше заданного порога. Если кластер похож на некоторые другие кластеры, существует вероятность того, что данные кластеры принадлежат одному человеку/транспортному средству. В этом случае система не закрепит такой кластер, чтобы иметь возможность провести повторную кластеризацию. Напротив, непохожие кластеры будут закреплены.
- **min_average_events_quality:** закрепить кластер, если среднее качество ассоциированных кластерных событий больше заданного минимального значения.

```
# pinned clusters keep their id and events after reclusterization
'CLUSTERS_AUTO_PIN_HEURISTICS': {
  'face': {
```

(continues on next page)

(продолжение с предыдущей страницы)

```

# pin clusters with `value` minimum cluster events
'min_events': {'enabled': True, 'value': 10},
# cluster's centroid similarity confidence is less then
'max_centroid_similarity_threshold': {'enabled': True, 'value': 0.54},
# minimum average event's quality
'min_average_events_quality': {'enabled': True, 'value': 0.45},
},
'body': {},
'car': {},
},

```

- Включите параметр PIN_MATCHED_CLUSTERS, чтобы закрепить кластеры и связанные с ними кластерные события, для которых есть совпадения в картотеке.

```

# always pin clusters with matched events (not affected by heuristics above)
'PIN_MATCHED_CLUSTERS': True,

```

10. При необходимости укажите максимальное количество кластерных событий в кластерах, которые являются «незакрепленными». По достижении данного количества кластеризация по расписанию будет автоматически отключена.

```

# skip clusterization if unpinned cluster events count is greater than this value
'CLUSTERIZATION_MAX_CLUSTER_EVENTS': None,

```

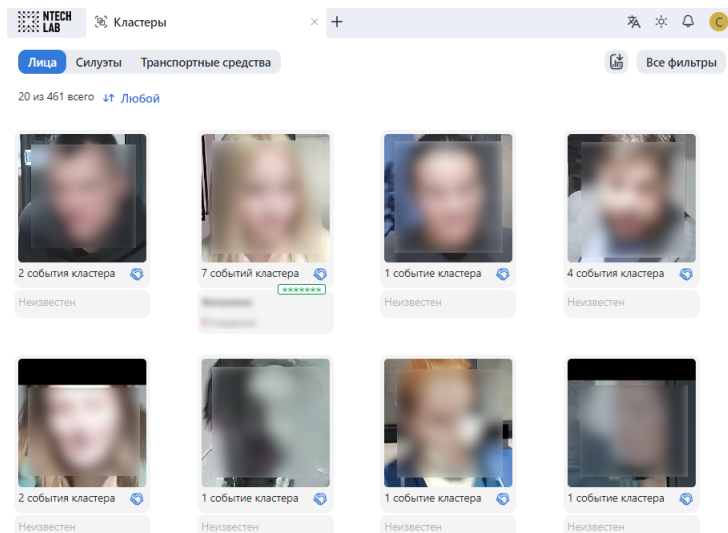
11. Перезапустите контейнер findface-multi-findface-multi-legacy-1. В веб-интерфейсе FindFace Multi появится вкладка *Кластеры*.

```
sudo docker container restart findface-multi-findface-multi-legacy-1
```

3.9.3 Работа с галереями кластеров

Просмотр кластеров. Фильтры

Для того чтобы просмотреть галереи кластеров, перейдите на вкладку *Кластеры*.



При работе с галереями кластеров используйте следующие фильтры:

Примечание: Некоторые фильтры из приведенного ниже списка могут быть скрыты, в зависимости от активированной функциональности распознавания.

- *Объект*: отобразить кластеры только для лиц, силуэтов или транспортных средств.
- *Совпадения*: отобразить кластеры с совпадением/без совпадения или любые.
- *Списки наблюдения*: отобразить только кластеры по определенному списку наблюдения.
- *Группы камер*: отобразить только кластеры по определенной группе камер.
- *Камеры*: отобразить только кластеры по определенной камере.
- *Имя карточки*: отобразить только кластеры по определенной карточке.
- *Дата и время*: отобразить только кластеры, сформированные в определенный период времени.
- *Первое событие кластера*: отобразить только первое событие кластера, сформированное в определенный период времени.
- *Событие кластера*: отобразить только события кластеров, сформированные в определенный период времени.
- *ID*: отобразить кластер с определенным ID.

Специальные фильтры для кластеров лиц

- *Возраст*: отобразить кластеры с людьми определенного возраста.
- *Борода*: фильтровать кластеры по наличию бороды.
- *Эмоции*: отобразить кластеры с заданными эмоциями.
- *Пол*: отобразить кластеры с людьми заданного пола.
- *Очки*: фильтровать кластеры по наличию очков на лице.
- *Витальность*: фильтровать кластеры по liveness лица.
- *Медицинская маска*: фильтровать кластеры по наличию медицинской маски.

Специальные фильтры для кластеров силуэтов

- *Пол по силуэту*: отобразить кластеры с силуэтами людей заданного пола.
- *Возраст по силуэту*: отобразить кластеры с силуэтами людей определенного возраста.
- *Головной убор*: отобразить только кластеры с людьми в головном уборе заданного типа: шапка/шляпа/кепка, капюшон/платок, без головного убора.
- *Жилет*: отобразить только кластеры с людьми, одетыми в жилет заданного цвета.
- *Жилет. Точность распознавания*: отобразить только кластеры с людьми, одетыми в жилет с заданной точностью распознавания.
- *Каска*: отображать только кластеры с людьми, одетыми в каску заданного цвета.
- *Каска. Точность распознавания*: отображать только кластеры с людьми, одетыми в каску с заданной точностью распознавания.

- *Цвет верха одежды*: отобразить только кластеры с людьми, одетыми в одежду заданного цвета (верх).
- *Цвет низа одежды*: отобразить только кластеры с людьми, одетыми в одежду заданного цвета (низ).
- *Тип верха одежды*: отобразить только те кластеры, в которых человек носит одежду верха заданного типа: куртка, пальто, безрукавка, толстовка, футболка, рубашка, платье.
- *Низ одежды*: отобразить только те кластеры, в которых человек носит одежду низа заданного типа: брюки, юбка, шорты, неопределенный.
- *Верх одежды*: отобразить только те кластеры, в которых человек носит одежду верха заданной обобщенной категории: с длинными рукавами, с короткими рукавами, без рукавов.

Специальные фильтры для кластеров транспортных средств

- *Марка*: фильтровать кластеры по марке транспортного средства.
- *Модель*: фильтровать кластеры по модели транспортного средства.
- *Тип кузова*: отобразить кластеры с транспортными средствами заданного типа кузова.
- *Цвет кузова*: отобразить кластеры с транспортным средством заданного цвета кузова.
- *Страна*: отобразить кластеры с транспортными средствами, зарегистрированными в заданной стране.
- *Регистрационный номер*: найти транспортное средство с заданным номером.
- *Регион*: отобразить кластеры с транспортными средствами, зарегистрированными в заданном регионе.
- *Цвет номера*: отобразить кластеры с заданным цветом регистрационного номера транспортного средства.
- *Спецтранспорт*: отображать только кластеры с транспортными средствами, принадлежащими заданному типу: полиция, автомобили МЧС и пожарные машины, газоспасательные и аварийно-спасательные службы, скорая помощь, военная техника, коммунальная, строительная техника, прочие.
- *Категория ТС*: отображать только кластеры с транспортными средствами, принадлежащими заданной категории: мотоцикл, скутер, легковой автомобиль с прицепом, грузовик, грузовик с прицепом, автобус, сочлененный автобус прочие.
- *Вес и размер кузова*: отобразить кластеры с транспортными средствами заданного веса и размера кузова.

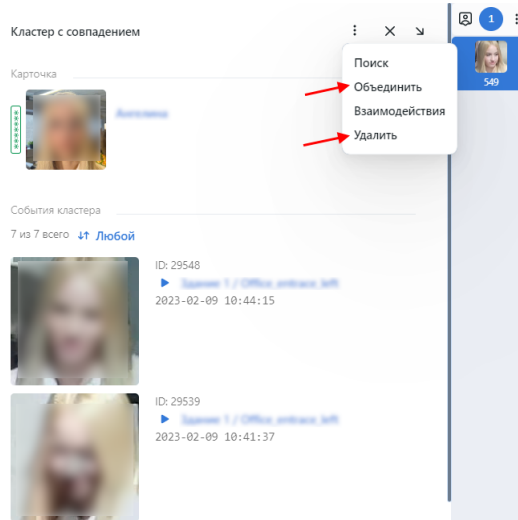
Примечание: Регион и цвет регистрационного знака определяются только для Объединенных Арабских Эмиратов (ОАЭ). Для других стран значения этих атрибутов будут выводиться как неизвестные в результатах распознавания.

Щелкните по нужному кластеру, чтобы просмотреть связанные с ним кластерные события. Вы будете перенаправлены на страницу *События кластера*.

Объединение и удаление кластеров

Для того чтобы вручную объединить несколько кластеров, выберите их один за другим и нажмите *Объединить*.

Для того чтобы удалить кластер, выберите его и нажмите *Удалить*.



3.9.4 Статическая галерея кластеров

Иногда бывает необходимо завершить кластеризацию объектов в определенный момент времени, а затем оперировать статической галереей сформированных кластеров.

Для отображения вкладки *Кластеры* при отключенной кластеризации выполните следующие действия:

1. Откройте файл конфигурации `/opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py`.

```
sudo vi /opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py
```

2. В раздел `SERVICES` вручную добавьте строку `"clusters": True`, как показано в примере ниже:

```
...
SERVICES = {
    "ffsecurity": {
        ...
        "clusters": True,
    }
    ...
}
```

3. Отключите процессы кластеризации в реальном времени и по расписанию.

```
...
'ENABLE_NIGHT_CLUSTERIZATION': False,
'ENABLE_REALTIME_CLUSTERIZATION': False,
```

4. Перезапустите контейнер `findface-multi-findface-multi-legacy-1`.

```
sudo docker container restart findface-multi-findface-multi-legacy-1
```

3.9.5 Ручная кластеризация

Для того чтобы вручную запустить процесс кластеризации, используйте утилиту `run_clusterization`.

Вы можете вызвать справку по опциям `run_clusterization`, выполнив следующую команду:

```
sudo docker -it exec findface-multi-findface-multi-legacy-1 /opt/findface-security/bin/
python3 /tigre_prototype/manage.py run_clusterization --help

usage: manage.py run_clusterization [-h]
                                   [--object-types OBJECT_TYPES [OBJECT_TYPES ...]]
                                   [--force] [--configuration CONFIGURATION]
                                   [--version] [-v {0,1,2,3}]
                                   [--settings SETTINGS]
                                   [--pythonpath PYTHONPATH] [--traceback]
                                   [--no-color] [--force-color]
                                   [--skip-checks]

optional arguments:
-h, --help            show this help message and exit
--object-types OBJECT_TYPES [OBJECT_TYPES ...]
                    Clusterize selected object types. Uses
                    CLUSTERIZE_OBJECT_TYPES from config if not provided.
                    Allowed types: face, body, car
--force              Force clusterization even if
                    CLUSTERIZATION_MAX_CLUSTER_EVENTS condition is met
--configuration CONFIGURATION
                    The name of the configuration class to load, e.g.
                    "Development". If this isn't provided, the
                    DJANGO_CONFIGURATION environment variable will be
                    used.
--version            show program's version number and exit
-v {0,1,2,3}, --verbosity {0,1,2,3}
                    Verbosity level; 0=minimal output, 1=normal output,
                    2=verbose output, 3=very verbose output
--settings SETTINGS The Python path to a settings module, e.g.
                    "myproject.settings.main". If this isn't provided, the
                    DJANGO_SETTINGS_MODULE environment variable will be
                    used.
--pythonpath PYTHONPATH
                    A directory to add to the Python path, e.g.
                    "/home/djangoprojects/myproject".
--traceback          Raise on CommandError exceptions
--no-color           Don't colorize the command output.
--force-color        Force colorization of the command output.
--skip-checks        Skip system checks.
```

Данная утилита позволяет по отдельности запускать кластеризацию лиц, силуэтов и транспортных средств, а также выполнять принудительную кластеризацию, если максимальное количество событий кластера превышает значение параметра `CLUSTERIZATION_MAX_CLUSTER_EVENTS` (см. *Включение и*

настройка кластеризации). Например, чтобы принудительно запустить кластеризацию лиц, выполните следующую команду:

```
sudo docker exec -it findface-multi-findface-multi-legacy-1 /opt/findface-security/bin/  
python3 /tigre_prototype/manage.py run_clusterization --object-types face --force
```

См.также:

- *Карточка*
- *Анализ взаимодействий*
- *Анализ аудитории*

3.10 Отчеты

В этой главе:

- *Типы отчетов*
- *Построение стандартного отчета*
- *Построение отчета об учете рабочего времени*
- *Работа с отчетами*

3.10.1 Типы отчетов

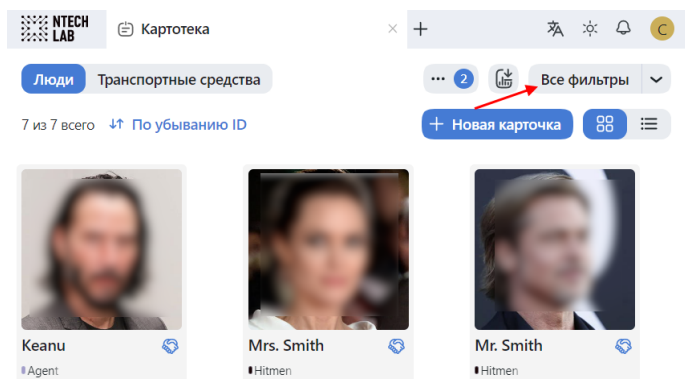
В FindFace Multi доступны следующие отчеты:

- Стандартные отчеты по следующим системным объектам:
 - *события*
 - *эпизоды*
 - *поиск*
 - *кластеры*
 - *источники видео*
 - *карточки*
 - *аналитические данные*
 - *журнал действий пользователей*
- Отчеты об учете рабочего времени, которые позволяют зафиксировать точные моменты входа и выхода сотрудников из предприятия или заданной области и подсчитать общее время, проведенное в данном месте, могут быть построены на вкладке *События*.

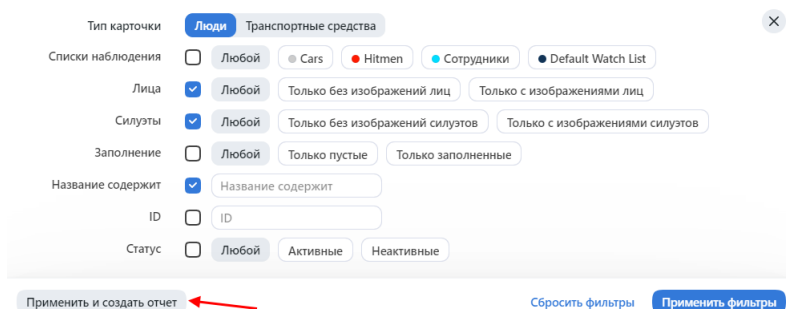
3.10.2 Построение стандартного отчета

Для того чтобы построить стандартный отчет по системному объекту, выполните следующие действия:

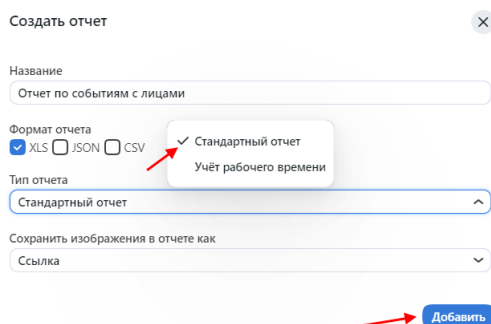
1. Перейдите на вкладку, соответствующую нужному системному объекту: *Поиск*, *Эпизоды и события*, *Кластеры*, *Источники видео*, *Картотека*, *Анализ аудитории*, *Журнал действий*.
2. Нажмите на кнопку *Все фильтры*. Установите фильтры для отчета.



3. Нажмите *Применить и создать отчет*.



4. Укажите имя отчета.
5. Если вы находитесь на вкладке *События*, выберите тип отчета *Стандартный отчет* (поскольку на данных вкладках доступно два типа отчетов).
6. Если применимо, выберите, следует ли сохранять изображения отчета в виде ссылок, миниатюр или полных кадров.
7. Нажмите *Добавить*. Отчет будет доступен для загрузки на вкладке *Отчеты*.



3.10.3 Построение отчета об учете рабочего времени

Отчет об учете рабочего времени можно построить на вкладке: *События*. Требуется указание камер входа и выхода.

Для того чтобы построить отчет об учете рабочего времени, выполните следующие действия:

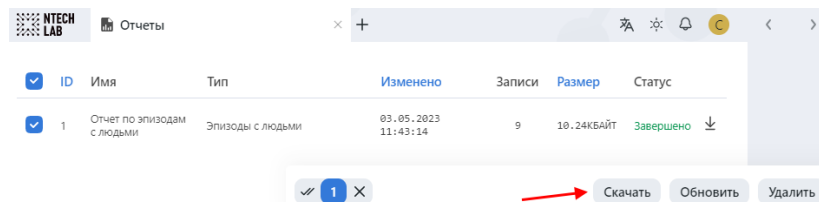
1. Установите фильтры для отчета.
2. Нажмите *Применить и создать отчет*.

3. Укажите имя отчета.
4. Выберите формат отчета.
5. Выберите тип отчета: *Учет рабочего времени*.
6. Если применимо, выберите, следует ли сохранять изображения отчета в виде ссылок, миниатюр или полных кадров.
7. Выберите камеру входа.
8. Выберите камеру выхода.
9. Нажмите *Добавить*. Отчет будет доступен для загрузки на вкладке *Отчеты*.

3.10.4 Работа с отчетами

Вы можете получить доступ к отчетам, ранее созданным в системе, на вкладке *Отчеты*. Доступны следующие операции:

- Скачивание выбранных отчетов.
- Обновление выбранных отчетов.
- Удаление выбранных отчетов.



См.также:

Настройка сохранения изображений в отчетах

3.11 Журнал действий пользователей

Комплексный журнал действий с возможностью поиска является отличным дополнительным инструментом для управления пользователями, обеспечивающим подробный аудит их действий и повышающим защиту системы. Вы можете получить доступ к этой функции на вкладке *Журнал действий*.

Пользователь	IP	ID устройства	Действие	Объект	ID объекта	Время
admin	192.168.1.1	123456789	Редактировать	Пользователь	1	07.03.2023 15:08:35
admin	192.168.1.1	123456789	Редактировать	Пользователь	1	07.03.2023 14:33:07
admin	192.168.1.1	123456789	Редактировать	Пользователь	1	07.03.2023 14:29:54
admin	192.168.1.1	123456789	Редактировать	Пользователь	1	07.03.2023 14:05:54

Каждая запись журнала содержит следующие данные:

- логин пользователя, выполнившего действие
- IP-адрес, с которого поступил запрос на выполнение действия
- ID устройства: уникальный идентификатор клиентского устройства пользователя
- тип действия, например, авторизация, поиск, изменение объекта, перезагрузка и т. д.
- тип объекта, к которому было приложено действие, например, карточка
- идентификатор объекта
- подробности в зависимости от типа действия
- временная метка

Используйте панель фильтров сверху, чтобы задать условия поиска в журнале.

3.12 Связанная с людьми аналитика

FindFace Multi является идеальным инструментом для сбора связанной с людьми аналитики. Включите формирование *кластеров* лиц и силуэтов и используйте их для извлечения значимой информации посредством набора аналитических функций.

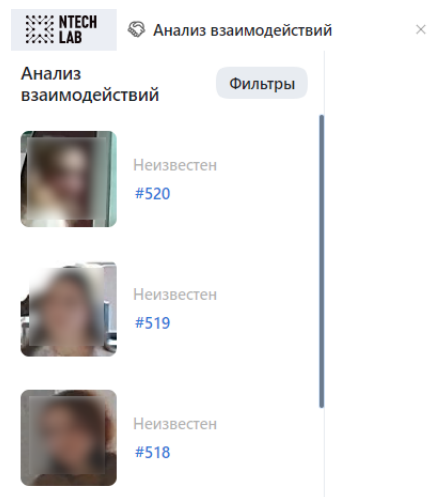
3.12.1 Анализ взаимодействий

FindFace Multi может выполнять отслеживание и анализ взаимодействий. Для использования данной функциональности вам потребуется включить *кластеризацию лиц*.

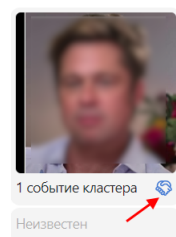
При отслеживании взаимодействий система автоматически определяет круг людей, с которыми человек ранее контактировал. Для каждого человека из первого круга система строит еще один круг ассоциированных людей и так далее. В целом, анализ взаимодействий состоит из трех кругов.

Примечание: Вы также можете отслеживать взаимодействия между силуэтами и транспортными средствами. Данная функция доступна через *HTTP API*. Обратите внимание, что кластеризация силуэтов и транспортных средств должна быть предварительно включена.

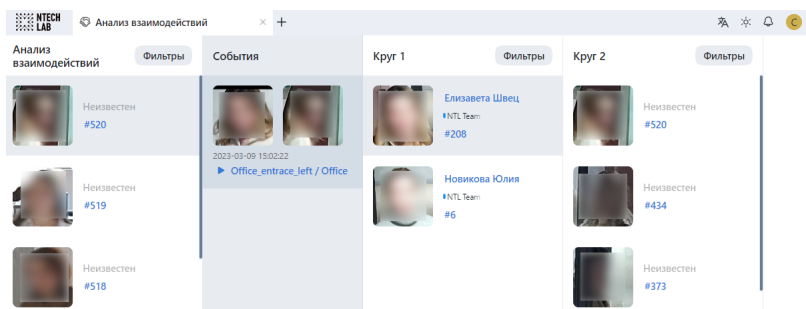
Анализ взаимодействий доступен на вкладке *Анализ взаимодействий*.



Совет: Вы также можете отобразить круг связанных людей непосредственно из вкладки *Кластеры*, нажав на значок рукопожатия.



На вкладке *Анализ взаимодействий* нажмите на изображение человека, чтобы отобразить первый круг его взаимодействий и ассоциированные события. Продолжайте, пока не раскроете все дерево взаимодействий.



Фильтры можно применять к каждому кругу:

Совет: Например, вы можете найти всех пожилых людей или людей без маски, кто прямо или косвенно взаимодействовал с потенциально инфицированным человеком.

- *Совпадения*: отобразить кластеры с совпадением/без совпадения или любые.
- *Списки наблюдения*: отобразить только кластеры по определенному списку наблюдения.
- *Группы камер*: отобразить только кластеры по определенной группе камер.
- *Камеры*: отобразить только кластеры по определенной камере.
- *Имя карточки*: отобразить только кластеры по определенной карточке.
- *Дата и время*: отобразить только кластеры, сформированные в определенный период времени.
- *Первое событие кластера*: отобразить только первое событие кластера, сформированное в определенный период времени.
- *Событие кластера*: отобразить только события кластеров, сформированные в определенный период времени.
- *ID*: отобразить кластер с определенным ID.

Специальные фильтры для кластеров лиц

- *Возраст*: отобразить кластеры с людьми определенного возраста.
- *Борода*: фильтровать кластеры по наличию бороды.
- *Эмоции*: отобразить кластеры с заданными эмоциями.
- *Пол*: отобразить кластеры с людьми заданного пола.
- *Очки*: фильтровать кластеры по наличию очков на лице.
- *Витальность*: фильтровать кластеры по liveness лица.
- *Медицинская маска*: фильтровать кластеры по наличию медицинской маски.

При поиске по кругу связей применяются следующие настройки:

- *Поиск связей по последнему событию эпизода*: использовать последнее событие эпизода для анализа контактов между людьми. В этом случае наиболее вероятно обнаружение действительно

связанных между собой людей, поскольку они одновременно покидают поле зрения камеры. Если опция отключена, система для поиска связей будет использовать лучшее событие эпизода.

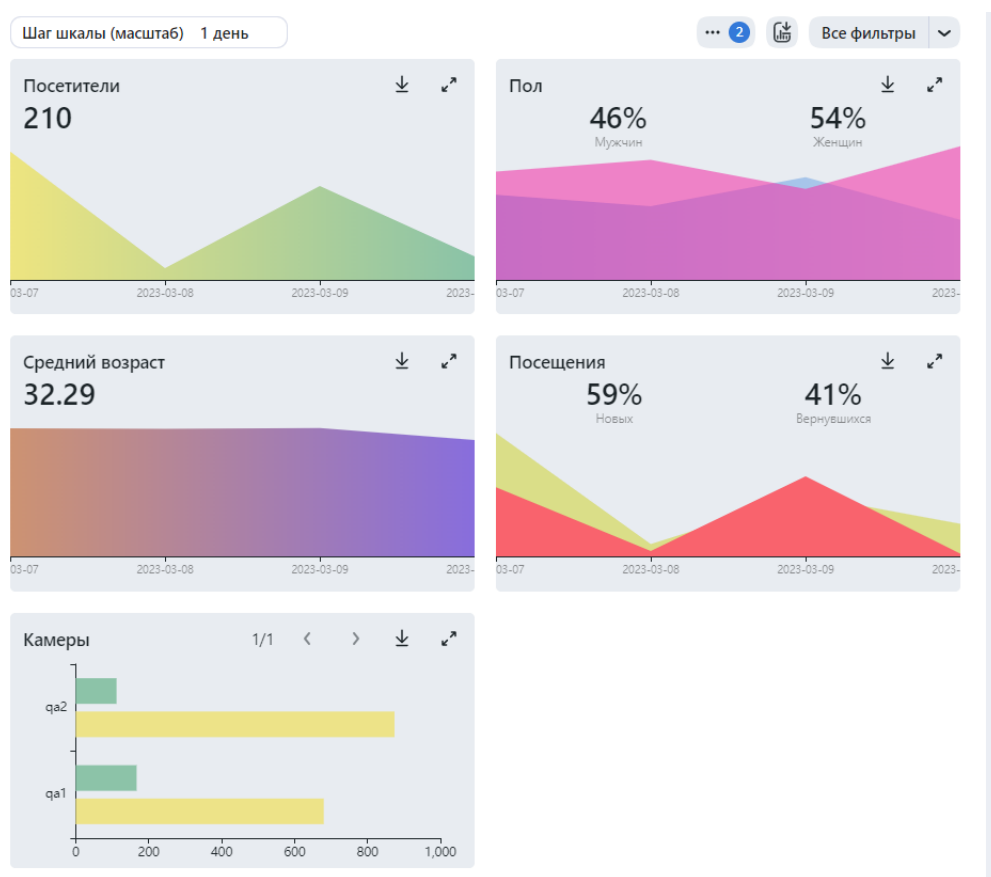
- *Максимальный промежуток между появлениями людей, чтобы считать их связанными, секунды*: максимальное время в секундах между появлением в кадре людей, чтобы считать их связанными.

3.12.2 Анализ аудитории

FindFace Multi включает статистику по количеству посетителей, их полу, среднему возрасту, наиболее часто посещаемым зонам (судя по самым активным камерам в системе) и характеру посещений (первое посещение или возвращение).

Диаграммы с аналитическими данными доступны на вкладке *Анализ аудитории*.

Важно: Построение аналитики выполняется только при включенной *кластеризации лиц*.



Для работы с данными аналитики используйте следующие фильтры:

- Дата и время
- Шаг шкалы
- Минимальное время между посещениями
- Количество возвращений
- Камеры

- Списки наблюдения
- Возраст
- Пол

См.также:

- *Кластеры лиц, силуэтов, транспортных средств*

3.13 Видеонаблюдение

В этой главе:

- *Видеостена*
 - *Отображение видео*
- *Видеомагнитофон*
 - *Включение видеозаписи с камеры*
 - *Просмотр видео с камеры в видеоплеере*

3.13.1 Видеостена

В FindFace Multi встроена базовая функциональность видеонаблюдения. Используйте видеостену для отображения видео с камер и видеофайлов.

Важно: Расширенные возможности видеонаблюдения доступны с помощью *Видеомагнитофона*.

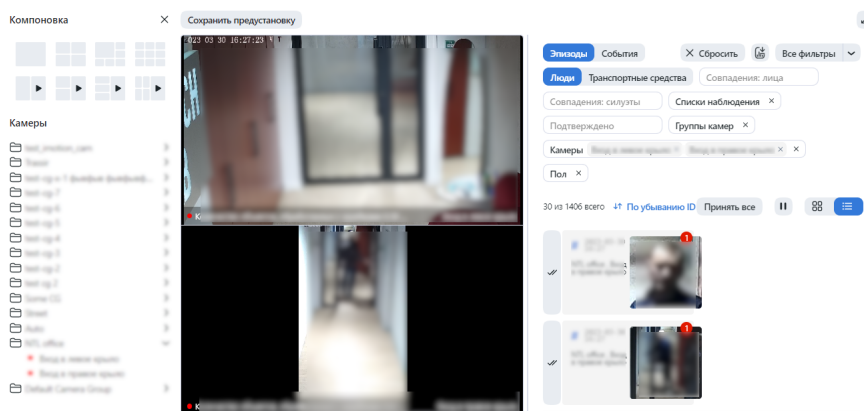
Отображение видео

Видеостена может работать в двух режимах (по четыре раскладки в каждом):

- видеотрансляция
- видеотрансляция с детектированием объектов и лентой эпизодов

Для отображения на видеостене видеоизображения выполните следующие действия:

1. Перейдите на вкладку *Видеостена*.
2. Выберите режим работы и раскладку камер.



3. Перетащите на видеостену выбранные камеры.

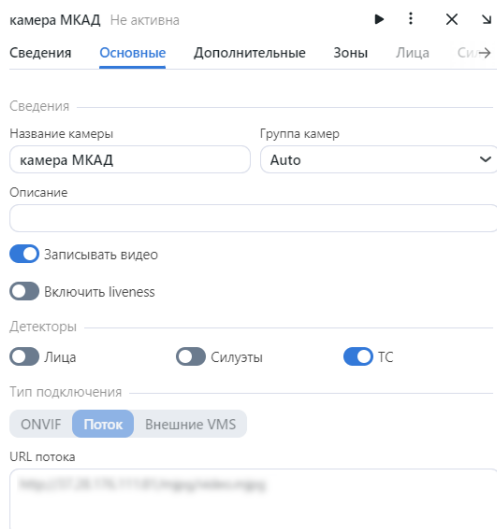
Вы можете работать с лентой эпизодов и событий на видеостене *по аналогии* с вкладкой *Эпизоды и события*, включая те же фильтры, которые доступны для эпизодов и событий.

Примечание: См. полный список *фильтров эпизодов* и *фильтров событий*, применимых к видеостене.

3.13.2 Видеомегнитофон

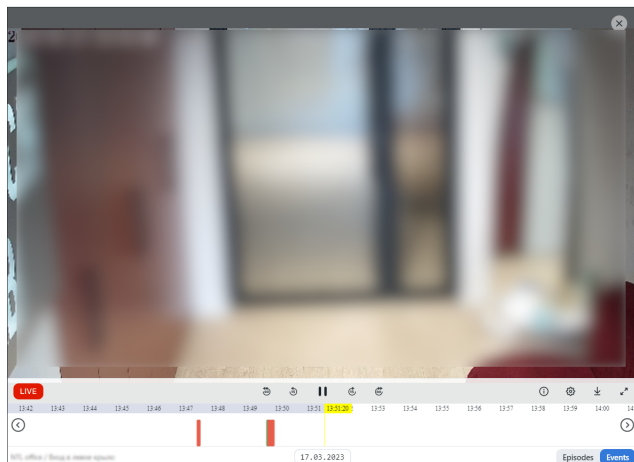
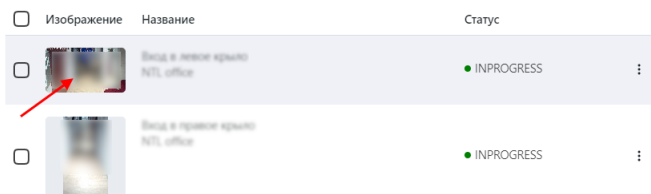
Включение видеозаписи с камеры

Если Видеомегнитофон *развернут* и *настроен*, в основных настройках камер появится флажок *Записывать видео*. Установите этот флажок, чтобы включить запись видеофрагментов с камеры и отправку их в Видеомегнитофон для дальнейшей обработки.



Просмотр видео с камеры в видеоплеере

При *определенных настройках* при нажатии на предварительный просмотр камеры вместо статического кадра открывается видеоплеер, транслирующий видео с камеры в режиме реального времени.



Видеоплеер имеет интуитивно понятный дизайн. Предоставляются следующие возможности:

1. Прямая трансляция с камеры.
2. Просмотр записанных с камеры видеофрагментов. Фрагменты будут отмечены на временной шкале фиолетовым цветом. Чтобы быстро переключиться с просмотра архивного видео на прямую трансляцию с камеры, нажмите кнопку *Live*.
3. Визуальная индикация моментов, соответствующих событиям с лицами или транспортными средствами (если данная функциональность активна), в виде цветных маркеров на временной шкале. События без совпадений отмечаются розовым цветом, а с совпадениями — салатовым.

Примечание: Данная функциональность является опциональной и должна быть предварительно настроена. См. [Настройка Видеомагнитофона](#).

Важно: Время на временной шкале отображается в часовом поясе зрителя. Например, если событие произошло в 14:00 в Абу-Даби, для зрителя в Париже его метка будет стоять на 12:00.

4. Перемещение в прошлое и будущее по временной шкале с возможностью увеличения и уменьшения ее масштаба.

Совет: По временной шкале можно перемещаться с помощью кнопок *< / >* или водить курсором по временной шкале с зажатой правой кнопкой мыши.

Совет: Для увеличения и уменьшения масштаба используйте колесико мыши с нажатой клавишей Ctrl.

5. Для экспорта выбранных клипов, выберите интервал и нажмите *Скачать*.



Руководство по интеграции

Данная глава посвящена возможностям интеграции с FindFace. В текущей версии для интеграции своей системы можно использовать только HTTP API.

4.1 HTTP API

Подробная интерактивная документация HTTP API FindFace доступна после установки по адресу <http://<findface-ip:port>/api-docs>. Изучайте и пробуйте.

Совет: Замените `<findface-ip:port>` на свой IP адрес FindFace Multi.

Совет: Документацию также можно найти в веб-интерфейсе, перейдя в меню по пунктам *Настройки* -> *Документация API*.

4.1.1 Обзор

Совет: Если после изучения данного раздела у вас останутся вопросы, не стесняйтесь обращаться к нашим специалистам по адресу support@ntechlab.com.

Аутентификация

Все методы API требуют простой HTTP-аутентификации на основе токенов. Для аутентификации введите слово «Token» и значение токена, разделенные пробелом, в HTTP-заголовок авторизации: `Authorization: Token 000....` Все запросы, которые не могут предоставить действительный токен аутентификации, приведут к ошибке авторизации HTTP 401.

Формат параметров

Для отправки запросов методами API поддерживаются следующие два формата:

- `application/json` : параметры тела запроса представляют собой JSON.
- `multipart/form-data`: параметры передаются составными частями, при этом загрузка фотоизображения осуществляется в том же запросе.

Дополнительная информация

Ограничения к вводным данным:

- Поддерживаемый формат изображений: JPEG, PNG, WEBP.
- Максимальный размер фотоизображения: 10 Мб.
- Максимальное разрешение для фотоизображения: 6000 пикселей на самой большой стороне.
- Минимальный размер лица: 50x50 пикселей.

Для осуществления ручной настройки см. `/opt/findface-multi/configs/findface-extraction-api/findface-extraction-api.yaml`.

4.1.2 Приступая к работе

В этой главе вы найдете рекомендуемую последовательность шагов, следуя которой вы сможете задействовать весь спектр функциональных возможностей FindFace Multi.

В этой главе:

- *Подготовка к работе*
- *Создание списков наблюдения и картотеки*
- *Загрузка и обработка видеофайла*
- *Источники данных*
- *FindFace Multi в действии*

Совет: Замените `<findface-ip:port>` на ваш IP адрес FindFace Multi, а `<token>` – на значение вашего токена.

Подготовка к работе

Выполните первичную настройку системы:

1. Выполните *развертывание FindFace Multi* и изучите API документацию <http://<findface-ip:port>/api-docs>.
2. Выполните *базовую HTTP авторизацию*, отправив логин и пароль в форму для ввода.
3. *Создайте свой токен авторизации* с вашим UUID сессии и опционально авторизацией по лицу.
4. Вставьте ваш токен следующим образом: Token <token> в поле API Key формы для авторизации или в HTTP заголовок Authorization.

Создание списков наблюдения и картотеки

1. *Создайте новый список наблюдения*, с помощью POST метода, или используйте список по умолчанию.
2. *Найдите ID идентификатор списка наблюдения* для того, чтобы отправить POST запрос на *создание новой карточки*.
3. Выполните *обнаружение лица (машины, силуэта)* на приложенном фотоизображении и скопируйте ID интересующего объекта..
4. Прикрепите фотографию *лица человека* с помощью `detection id`.

Загрузка и обработка видеофайла

1. Организуйте видеонаблюдение с помощью *загрузки видеофайла*.
2. Найдите `id` из *списка видеоархивов* или путем *добавления нового видеоархива*.
3. *Прикрепите исходный видеофайл* вместе с `id` видеоархива.
4. *Запустите обработку видеоархива*.

Источники данных

Для настройки видеомониторинга объектов добавьте камеры в FindFace Multi, сгруппировав их с учетом расположения.

1. В системе доступна группа камер по умолчанию. *Список доступных групп камер*.
2. При необходимости *создайте новую группу камер*.
3. Используйте `id` группы камер, чтобы отправить POST запрос для *создания новой камеры*.

FindFace Multi в действии

1. Ищите объекты в базе обнаруженных объектов и картотеке. *Подробнее.*
2. *Сравнивайте объекты*, чтобы проверить их на совпадение.

4.1.3 Аутентификация

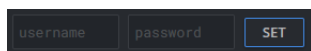
Все API запросы необходимо отправлять на адрес `http://<findface-ip:port>`.

Аутентификация HTTP Basic

Базовая аутентификация HTTP представляет собой простой механизм аутентификации «запрос-ответ», который может использоваться сервером для запроса аутентификационной информации (имя пользователя и пароль).

Введите свои учетные данные в форму для входа в систему:

- Username – имя пользователя вашей учетной записи FindFace Multi.
- Password – пароль для вашей учетной записи FindFace Multi.



Эти данные должны быть переданы в заголовке запроса `Authorization` в виде base64 кода.

```
Authorization: Basic <base64(username:password)>
```

Авторизация API Key

Для того чтобы создать токен авторизации, используйте следующий метод:

```
POST /auth/login/
```

Предварительно вы должны предоставить основные учетные данные для авторизации (имя пользователя, пароль).

Этот метод возвращает в ответе токен авторизации, который вы можете использовать в остальных API методах.

Если система использует авторизацию по лицу и паролю, то вместе с заголовком базовой авторизации дополнительно передайте токен, полученный во время авторизации по лицу, в поле `video_auth_token`.

Тело запроса является обязательным и содержит объект в формате `application/json` со следующими параметрами:

Совет: * - обозначает обязательные параметры. – только для чтения.

Параметры	Схема JSON	Описание
video_auth_token	string	Ограничения: Минимум 1 символ. Неактивный токен при авторизации лицом. Требуется, если задано <i>face_and_password</i> .
uuid*	string	Ограничения: от 1 до 256 символов. Уникальный идентификатор сессии на устройстве.
mobile	boolean	Устройство мобильное.
device_info	<any-key>: stringfloatbooleanobjectarraynull	Информация об устройстве.

Пример запроса

```
{
  "video_auth_token": "A",
  "uuid": "A",
  "mobile": false,
  "device_info": {}
}
```

CURL пример

```
curl -X POST "http://<findface-ip:port>/auth/login/" \
-H "Accept: application/json" \
-H "Content-Language: ru" \
-H "Accept-Language: ru" \
-H "Authorization: Basic <base64(username:password)>" \
-H "Content-Type: application/json" \
-d '{"uuid":"A"}' \
```

Возвращает:

- 200 – в случае успеха.
- 400 – если тело запроса не определено.

В случае успешного ответа (ОК: 200), возвращается объект со следующими *параметрами*. *Пример*.

Пример ответа

Параметры	Схема JSON	Описание
token*	string	Токен авторизации.
user*	{...}	Информация о пользователе.
token_expiration_datetime	datetime	Дата и время истечения срока действия токена.

user содержит следующие параметры:

- `id* : integer`
- `active* : boolean`
- `created_date* : date-time`
- `groups : [integer]`
- `modified_date* : date-time` – Дата и время изменения объекта.
- `permissions* : [string]`
- `real_name* : string` – Ограничения: максимум 80 символов.
- `name* : string` – Максимум 80 символов.
- `comment : string` – Расширенное описание пользователя, до 2048 символов.
- `camera_group_permissions* : {<any-key>: string}`
- `watch_list_permissions* : {<any-key>: string}`
- `group_permissions* : {<any-key>: string}`
- `primary_group* : integer`
- `language : enum` – Допустимые значения: `en-usesru`. Язык пользовательского интерфейса.
- `has_face* : boolean` – Фотография пользователя предоставлена.
- `face_cover : stringnull` – Ограничения: Максимум 32 символа.
- `ad_user* : boolean` – Пользователь зарегистрирован в Active Directory.

Пример ответа

```
{
  "token": "string",
  "user": {
    "id": 0,
    "active": false,
    "created_date": "1970-01-01T00:00:00.000Z",
    "groups": [
      0
    ],
    "modified_date": "1970-01-01T00:00:00.000Z",
    "permissions": [
      "string"
    ],
    "real_name": "AAAAAA",
    "name": "AAAAAA",
    "comment": "AAAAAA",
    "camera_group_permissions": {},
    "watch_list_permissions": {},
    "group_permissions": {},
    "primary_group": 0,
    "language": "en-us",
    "has_face": false,
    "face_cover": "AAAAAA",
    "ad_user": false
  }
}
```

(continues on next page)

(продолжение с предыдущей страницы)

```
},
"token_expiration_datetime": "1970-01-01T00:00:00.000Z"
}
```

Вставьте полученный токен в поле **API Key** формы для авторизации: **Token <token>** и нажмите **SET** или отправьте в заголовке авторизации: **Authorization: Token 000...** Все запросы, которые не могут предоставить действительный токен аутентификации, приведут к ошибке авторизации HTTP 401.



Для того чтобы отключить примененные ключи, нажмите **REMOVE** или **CLEAR ALL API KEYS** или отправьте запрос **POST /auth/logout/**.

4.1.4 Списки наблюдения

Отобразить списки наблюдений

Для того чтобы отобразить списки наблюдений, используйте следующий метод:

```
GET /watch-lists/
```

Запрос содержит следующие параметры QUERY-STRING PARAMETERS:

Название	Схема JSON	Описание
limit	integer	Количество возвращаемых результатов.
ordering	string	Допустимые значения: id, created_date, modified_date.

Для того чтобы найти ID списка наблюдения, укажите произвольное значение для **limit** и **ordering**.

CURL пример

```
curl -X GET "http://<findface-ip:port>/watch-lists/?limit=2&ordering=id" \
-H "Accept: application/json" \
-H "Content-Language: ru" \
-H "Accept-Language: ru" \
-H "Authorization: Token <token>" \
```

В случае успешного ответа (ОК: 200) возвращается объект, содержащий массив **results** со *следующими параметрами*. *Пример*.

Совет: * – означает, что параметр является обязательным. – только для чтения.

Название	Схема JSON	Описание
id*	integer	ID списка наблюдения.
created_date*	date-time	Дата создания объекта.
modified_date*	date-time	Дата изменения объекта.
active	boolean	true , если список наблюдения активен.
name*	string	Название списка наблюдения, до 256 знаков.
comment	string	Комментарий, до 2048 знаков.
color	string	Цвет метки объекта в шестнадцатеричном формате. Ограничения: Максимум 6 символов.
notify	boolean	true , если включены звуковые оповещения.
acknowledge	boolean	Требуется подтверждение событий, которые соответствуют списку наблюдения.
permissions	{<any-key>: string}	Разрешения.
camera_groups	[integer]	Группы камер, используемые в списке наблюдения.
face_threshold	numbernull	Порог схожести для распознавания лиц для текущего списка наблюдения (от 0 до 1).
body_threshold	numbernull	Порог схожести для распознавания силуэтов для текущего списка наблюдения (от 0 до 1).
car_threshold	numbernull	Порог схожести для распознавания автомобилей для текущего списка наблюдения (от 0 до 1).
ignore_events	boolean	События не будут создаваться, если установлено значение True .
send_events_to_external_sinks	boolean	true , если события, соответствующие этому списку наблюдения, должны быть отправлены на внешнюю виртуальную машину.
active_after	date-time	Информация о дате и времени.
active_before	date-time	Информация о дате и времени.
disable_schedule	{...}	Сериализатор <code>mixin</code> , который вызывает ошибку проверки <code>ValidationError</code> , если представлены избыточные поля. Может использоваться во вложенных сериализаторах.
recount_schedule_on*	date-time	Информация о дате и времени.
origin	string	Максимум 256 символов.

Пример ответа

```
{
  "results": [
    {
      "id": -1,
      "created_date": "2023-01-16T13:44:36.407610Z",
      "modified_date": "2023-01-16T13:44:36.407750Z",
      "active": true,
      "name": "Unmatched",
      "comment": "Default list for unmatched events",
      "color": "ffffff",
      "notify": false,
      "acknowledge": false,
```

(continues on next page)

(продолжение с предыдущей страницы)

```

    "permissions": {
      "1": "edit",
      "2": "view",
      "3": "view"
    },
    "camera_groups": [],
    "face_threshold": null,
    "body_threshold": null,
    "car_threshold": null,
    "ignore_events": false,
    "send_events_to_external_vms": false,
    "active_after": null,
    "active_before": null,
    "disable_schedule": {},
    "recount_schedule_on": null,
    "origin": "ffsecurity"
  },
  {
    "id": 1,
    "created_date": "2023-01-16T13:44:36.394542Z",
    "modified_date": "2023-01-16T13:44:36.394577Z",
    "active": true,
    "name": "Default Watch List",
    "comment": "",
    "color": "123456",
    "notify": false,
    "acknowledge": false,
    "permissions": {
      "1": "edit",
      "2": "view",
      "3": "view"
    },
    "camera_groups": [],
    "face_threshold": null,
    "body_threshold": null,
    "car_threshold": null,
    "ignore_events": false,
    "send_events_to_external_vms": false,
    "active_after": null,
    "active_before": null,
    "disable_schedule": {},
    "recount_schedule_on": null,
    "origin": "ffsecurity"
  }
]
}

```

Создать новый список наблюдений

Чтобы добавить новый список наблюдений, используйте следующий метод:

```
POST /watch-lists/
```

Тело запроса является обязательным и содержит объект application/json со следующими *параметрами списка наблюдения*.

Название	Схема JSON	Описание
active	boolean	true , если список наблюдения активен.
name*	string	Название списка наблюдения, до 256 знаков.
comment	string	Комментарий, до 2048 знаков.
color	string	Цвет метки объекта в шестнадцатеричном формате. Ограничения: от 1 до 6 символов.
notify	boolean	true , если включены звуковые оповещения.
acknowledge	boolean	Требуется подтверждение событий, которые соответствуют списку наблюдения.
permissions	<any-key>: string	Ограничения: Минимум 1 символ.
camera_groups	[integer]	Группы камер, используемые в списке наблюдения.
face_threshold	numbernull	Порог схожести для распознавания лиц для текущего списка наблюдения (от 0 до 1).
body_threshold	numbernull	Порог схожести для распознавания силуэтов для текущего списка наблюдения (от 0 до 1).
car_threshold	numbernull	Порог схожести для распознавания автомобилей для текущего списка наблюдения (от 0 до 1).
ignore_events	boolean	События не будут создаваться, если установлено значение True .
send_events_to_external_sims	boolean	true , если события, соответствующие этому списку наблюдения, должны быть отправлены на внешнюю виртуальную машину.
active_after	date-timenull	Информация о дате и времени.
active_before	date-timenull	Информация о дате и времени.
disable_schedule	{...}	Сериализатор mixin, который вызывает ошибку проверки ValidationError , если представлены избыточные поля. Может использоваться во вложенных сериализаторах.
origin	string	Ограничения: от 1 до 256 символов.

Пример запроса

Совет: Этот пример приведен только для ознакомления, подставьте свои значения в соответствующие поля. Вы можете заполнить только обязательные поля, а остальные будут по умолчанию.

```
{
  "active": false,
  "name": "A",
  "comment": "AAAAAA",
```

(continues on next page)

(продолжение с предыдущей страницы)

```

"color": "A",
"notify": false,
"acknowledge": false,
"permissions": {},
"camera_groups": [
  0
],
"face_threshold": 0,
"body_threshold": 0,
"car_threshold": 0,
"ignore_events": false,
"send_events_to_external_vms": false,
"active_after": "1970-01-01T00:00:00.000Z",
"active_before": "1970-01-01T00:00:00.000Z",
"disable_schedule": {
  "monday": [
    [
      "A"
    ]
  ],
  "tuesday": [
    [
      "A"
    ]
  ],
  "wednesday": [
    [
      "A"
    ]
  ],
  "thursday": [
    [
      "A"
    ]
  ],
  "friday": [
    [
      "A"
    ]
  ],
  "saturday": [
    [
      "A"
    ]
  ],
  "sunday": [
    [
      "A"
    ]
  ]
},
"origin": "A"

```

(continues on next page)

(продолжение с предыдущей страницы)

```
}
```

Вы можете отправить в запросе следующие параметры:

```
{
  "active": true,
  "name": "Test_list",
  "comment": "AAAAAA",
  "color": "35a2ee"
}
```

CURL пример

```
curl -X POST "http://<findface-ip:port>/watch-lists/" \
-H "Accept: application/json" \
-H "Content-Language: ru" \
-H "Accept-Language: ru" \
-H "Authorization: Token <token>" \
-H "Content-Type: application/json" \
-d '{"active":true,"name":"Test_list","comment":"AAAAAA","color":"35a2ee"}' \
```

Если ответ успешен (Created: 201), то возвращается объект, который содержит следующие *параметры*.
Пример.

Пример ответа

```
{
  "id": 2,
  "created_date": "2023-01-16T14:07:52.424520Z",
  "modified_date": "2023-01-16T14:07:52.424549Z",
  "active": true,
  "name": "Test_list",
  "comment": "AAAAAA",
  "color": "35a2ee",
  "notify": false,
  "acknowledge": false,
  "permissions": {
    "1": "edit"
  },
  "camera_groups": [],
  "face_threshold": 0,
  "body_threshold": 0,
  "car_threshold": 0,
  "ignore_events": false,
  "send_events_to_external_vms": false,
  "active_after": null,
  "active_before": null,
  "disable_schedule": {},
  "recount_schedule_on": null,
}
```

(continues on next page)

(продолжение с предыдущей страницы)

```
"origin": "ffsecurity"
}
```

Используйте `id` списка наблюдения, чтобы отправить `POST` запрос для *создания новой карточки*.

Полезные API запросы

```
GET /watch-lists/
POST /watch-lists/
GET /watch-lists/{id}/
DELETE /watch-lists/{id}/
PATCH /watch-lists/{id}/
POST /watch-lists/{id}/purge/
GET /watch-lists/count/
POST /watch-lists/purge_all/
```

4.1.5 Карточка

Создание новой карточки

Чтобы создать новую карточку, используйте следующий метод:

```
POST /cards/humans/
```

Совет: Для карточек ТС в запросе вместо `humans` укажите `cars`.

Тело запроса является обязательным и содержит объект `application/json` со следующими *параметрами*.

Совет: * – означает, что параметр является обязательным. – только для чтения.

Название	Схема JSON	Описание
<code>active</code>	<code>boolean</code>	<code>true</code> , если объект активен.
<code>name*</code>	<code>string</code>	Название карточки, до 256 символов.
<code>comment</code>	<code>string</code>	Комментарий, до 2048 символов.
<code>watch_lists*</code>	<code>[integer]</code>	Массив из ID списков наблюдений.
<code>meta</code>	Любой тип данных	Метаданные.
<code>active_after</code>	<code>date-timenum</code>	Информация о дате и времени.
<code>active_before</code>	<code>date-timenum</code>	Информация о дате и времени.
<code>disable_schedule</code>	<code>{...}</code>	Сериализатор <code>mixin</code> , который вызывает ошибку проверки <code>ValidationError</code> , если представлены избыточные поля. Может использоваться во вложенных сериализаторах.

Пример запроса

Совет: Этот пример приведен только для ознакомления, подставьте свои значения в соответствующие поля. Вы можете заполнить только обязательные поля, а остальные будут по умолчанию.

```
{
  "active": false,
  "name": "A",
  "comment": "AAAAAA",
  "watch_lists": [
    0
  ],
  "meta": {},
  "active_after": "1970-01-01T00:00:00.000Z",
  "active_before": "1970-01-01T00:00:00.000Z",
  "disable_schedule": {
    "monday": [
      [
        "A"
      ]
    ],
    "tuesday": [
      [
        "A"
      ]
    ],
    "wednesday": [
      [
        "A"
      ]
    ],
    "thursday": [
      [
        "A"
      ]
    ],
    "friday": [
      [
        "A"
      ]
    ],
    "saturday": [
      [
        "A"
      ]
    ],
    "sunday": [
      [
        "A"
      ]
    ]
  ]
}
```

(continues on next page)

(продолжение с предыдущей страницы)

```
}  
}
```

Вы можете отправить в запросе следующие параметры:

```
{  
  "active": true,  
  "name": "Angelina Jolie",  
  "comment": "Миссис Смит",  
  "watch_lists": [  
    2  
  ]  
}
```

CURL пример

```
curl -X POST "http://<findface-ip:port>/cards/humans/" \  
-H "Accept: application/json" \  
-H "Content-Language: ru" \  
-H "Accept-Language: ru" \  
-H "Authorization: Token <token>" \  
-H "Content-Type: application/json" \  
-d '{"active":true,"name":"Angelina Jolie","comment":"Миссис Смит","watch_lists":[2]}' \  

```

Если ответ успешен (Created: 201), то возвращается объект, который содержит следующие *параметры*.
Пример.

Название	Схема JSON	Описание
id*	integer	ID карточки.
active	boolean	true , если объект активен.
filled*	boolean	true , если карточка заполнена.
created_date*	date-time	Дата создания.
modified_date*	date-time	Дата изменения.
name*	string	Название карточки, до 256 символов.
comment	string	Комментарий, до 2048 символов.
watch_lists*	[integer]	Массив из ID списков наблюдений.
looks_like* <ul style="list-style-type: none"> confidence* collection* matched_object* 	<ul style="list-style-type: none"> number stringnull stringnull 	<ul style="list-style-type: none"> Порог схожести – от 0 до 1. Группа карточек. Совпадающий объект.
meta	Любой тип данных	Метаданные.
looks_like_confidence*	number	Степень схожести.
active_after	date-time null	Информация о времени и дате.
active_before	date-time null	Информация о времени и дате.
disable_schedule	массив	Сериализатор mixin, который вызывает ошибку проверки ValidationError , если представлены избыточные поля. Может использоваться во вложенных сериализаторах.
recount_schedule_on*	date-time	Информация о времени и дате.
face_objects*	integer	Счетчик связанных лиц.
body_objects*	integer	Счетчик связанных силуэтов.
face_cluster*	integer	Кластер лиц.
body_cluster*	integer	Кластер силуэтов.
links_to_relations* <ul style="list-style-type: none"> id* name* created_date* card* relation* 	<ul style="list-style-type: none"> integer string date-time integer integer 	<ul style="list-style-type: none"> ID Название, до 256 символов. Дата создания связи. Связанная карточка. Связь.

Примечание: Связи не поддерживаются в FindFace Multi 2.0 и хранятся только в API.

Пример ответа

```
{
  "id": 1,
  "active": true,
  "filled": true,
  "created_date": "2023-01-17T07:43:56.363330Z",
  "modified_date": "2023-01-17T07:43:56.363354Z",
  "name": "Angelina Jolie",
  "comment": "Миссис Смит",
  "watch_lists": [
    2
  ],
  "meta": {},
  "active_after": null,
  "active_before": null,
  "disable_schedule": {},
  "recount_schedule_on": null,
  "face_objects": 0,
  "body_objects": 0,
  "face_cluster": null,
  "body_cluster": null,
  "links_to_relations": []
}
```

Вывести список карточек

Чтобы вывести список карточки «Люди» с их параметрами, используйте следующий метод:

```
GET /cards/humans/
```

Запрос содержит параметры QUERY-STRING PARAMETERS. Подробнее <http://<findface-ip:port>/api-docs>.

Вы можете использовать этот метод для *поиска лиц* в системе. Укажите значение `detection:<detection id>`, полученное при *обнаружении объекта на фотографии*, в поле `looks_like`.

CURL пример

```
curl -X GET "http://<findface-ip:port>/cards/humans/?looks_like=detection
↪%3Acf4ffvmv54rotim9jt60" \
-H "Accept: application/json" \
-H "Content-Language: ru" \
-H "Accept-Language: ru" \
-H "Authorization: Token <token>" \
```

Если ответ успешен (OK: 200), то возвращается объект со следующими параметрами:

Название	Схема JSON	Описание
<code>next_page</code>	stringnull	Следующая страница.
<code>prev_page</code>	stringnull	Предыдущая страница.
<code>results</code>	[{...}]	Содержит <i>параметры</i> .

Пример ответа

```
{
  "next_page": null,
  "prev_page": null,
  "results": [
    {
      "id": 1,
      "active": true,
      "filled": true,
      "created_date": "2023-01-17T07:43:56.363330Z",
      "modified_date": "2023-01-17T11:56:42.496871Z",
      "name": "Angelina Jolie",
      "comment": "Миссис Смит",
      "watch_lists": [
        2
      ],
      "looks_like": {
        "confidence": 0.8078,
        "collection": "face_objects",
        "matched_object": "4493493039043981648"
      },
      "meta": {},
      "looks_like_confidence": 0.8078,
      "active_after": null,
      "active_before": null,
      "disable_schedule": {},
      "recount_schedule_on": null,
      "face_objects": 1,
      "body_objects": 0,
      "face_cluster": 11,
      "body_cluster": null,
      "links_to_relations": []
    }
  ]
}
```

Полезные API запросы

```
GET /cards/cars/
POST /cards/cars/
GET /cards/cars/{id}/
DELETE /cards/cars/{id}/
PATCH /cards/cars/{id}/
GET /cards/cars/count/
GET /cards/humans/
POST /cards/humans/
GET /cards/humans/{id}/
DELETE /cards/humans/{id}/
PATCH /cards/humans/{id}/
GET /cards/humans/count/
```

4.1.6 Обнаружение объектов на фотографии

Чтобы обнаружить объект на фотографии, используйте следующий метод:

```
POST /detect
```

Тело запроса является обязательным и содержит multipart/form-data со следующими параметрами.

Название	Схема JSON	Описание
photo	binary	Исходный файл изображения.
attributes	объект	Атрибуты для объектов: лицо, автомобиль и силуэт.

Поле **attributes** может быть пустым или содержать объекты **face**, **car** и **body** со следующими параметрами для каждого объекта:

- age (возраст): тип boolean
- beard (борода): тип boolean
- emotions (эмоции): тип boolean
- glasses (очки): тип boolean
- gender (пол): тип boolean
- medmask (медицинская маска): тип boolean
- headpose (наклон головы): тип boolean

Прикрепите исходный файл изображения и отправьте POST запрос.

CURL пример

```
curl -X POST "http://<findface-ip:port>/detect" \
-H "Accept: application/json" \
-H "Content-Language: ru" \
-H "Accept-Language: ru" \
-H "Authorization: Token <token>" \
-H "Content-Type: multipart/form-data" \
-F "photo=@Смит_01.png" \
-F "attributes={
  \"face\": {
    \"age\": false,
    \"beard\": false,
    \"emotions\": false,
    \"glasses\": false,
    \"gender\": false,
    \"medmask\": false,
    \"headpose\": false
  },
  \"car\": {
    \"description\": false,
    \"license_plate\": false,
    \"special_vehicle_type\": false,
    \"category\": false,
    \"weight_type\": false,
```

(continues on next page)

(продолжение с предыдущей страницы)

```

    "orientation": false
  },
  "body": {
    "color": false,
    "clothes": false,
    "bags": false,
    "protective_equipment": false,
    "age_gender": false
  }
} " \

```

В случае успешного ответа (OK: 200), возвращается объект со следующими *параметрами*. *Пример*.

Название	Схема JSON	Описание
orientation*	integer	Ориентация фотографии в формате EXIF.
objects	поддерживается любой из форматов: strintfloatboolobjectarraynull	Возвращаемые объекты с запрашиваемыми атрибутами.

Пример ответа

```

{
  "orientation": 1,
  "objects": {
    "face": [
      {
        "id": "cf0mbqev54rqhngnq940",
        "bbox": {
          "left": 451,
          "top": 235,
          "right": 645,
          "bottom": 502
        },
        "detection_score": 0.80645436,
        "low_quality": false,
        "features": {}
      },
      {
        "id": "cf0mbqev54rqhngnq94g",
        "bbox": {
          "left": 757,
          "top": 79,
          "right": 948,
          "bottom": 353
        },
        "detection_score": 0.90099674,

```

(continues on next page)

(продолжение с предыдущей страницы)

```
    "low_quality": false,  
    "features": {}  
  }  
]  
}  
}
```

В ответе вы получите ID объекта и координаты прямоугольника вокруг объекта (лица, автомобиля, силуэта). Скопируйте значение `id` интересующего вас объекта, чтобы использовать его для *добавления объекта* в карточку или для *поиска объекта*.

4.1.7 Объекты

Добавить объект «Лицо»

```
POST /objects/faces/
```

Данный метод позволяет создать объект «Лицо», который содержит исходное фотоизображение, миниатюру лица и другие атрибуты.

Совет: Для объектов «Автомобиль» и «Силуэт» в запросах используйте `cars` или `body` вместо `faces`.

Тело запроса является обязательным и содержит multipart/form-data со следующими *параметрами*:

Название	Схема JSON	Описание
<code>create_from</code>	string	Данное поле может содержать один из следующих вариантов идентификатора объекта: <ul style="list-style-type: none"> • detection:<detection id>: используйте обнаруженный объект с помощью метода <code>POST /detect</code> на исходном изображении. Исходное фотоизображение в поле <code>source_photo</code> должно совпадать с фотоизображением в запросе <code>POST /detect</code>. • event:<event id> для лица, силуэта или автомобиля: создавайте новый объект из События (поле <code>source_photo</code> должно быть пустым). • object:<object id> для лица, силуэта или автомобиля: используйте другой объект в качестве шаблона для текущего объекта (поле <code>source_photo</code> должно быть пустым).
<code>detect_id</code>	string	Вспомогательный параметр.
<code>mf_selector</code>	enum	Этот параметр определяет действие FindFace Multi, когда в <code>source_photo</code> присутствует несколько объектов, а <code>create_from</code> не задан. Значение по умолчанию: <code>reject</code> . Допустимые значения: <ul style="list-style-type: none"> • reject – Отклонить, если в исходном фотоизображении несколько объектов. • biggest – Выбрать наибольший объект в исходном фотоизображении.
<code>upload_list</code>	integer	Добавить объект в данный список загрузок.
<code>source_photo</code>	binary	Исходное фотоизображение (является обязательным, если в поле <code>create_from</code> указан <code>detection:<detection id></code> или остается пустым)
<code>frame_coords_left</code>	integer	Левая граница прямоугольника вокруг объекта.
<code>frame_coords_top</code>	integer	Верхняя граница прямоугольника вокруг объекта.
<code>frame_coords_right</code>	integer	Правая граница прямоугольника вокруг объекта.
<code>frame_coords_bottom</code>	integer	Нижняя граница прямоугольника вокруг объекта.
<code>active</code>	boolean	<code>true</code> , если объект активен. Значение по умолчанию <code>true</code> .
<code>card</code>	integer	ID связанной карточки.

Прикрепите фотоизображение в поле `source_photo`, укажите ID соответствующей карточки и вставьте `detection:<detection id>` в форму `create_from`, где `detection id` полученный `id` при *обнаружении объекта на фотоизображении*. В поле `source_photo` должно быть то же изображение, что и в запросе `POST /detect`.

create_from

string

detection:cf2g86uv54rqhngnq960

source_photo

binary

Выберите файл

Смит_01.png

card

integer

3

CURL пример

```
curl -X POST "http://<findface-ip:port>/objects/faces/" \
-H "Accept: application/json" \
-H "Content-Language: ru" \
-H "Accept-Language: ru" \
-H "Authorization: Token <token>" \
-H "Content-Type: multipart/form-data" \
-F "create_from=detection:cf2g86uv54rqhngnq960" \
-F "source_photo=@Смит_01.png" \
-F "card=3" \
```

В случае успешного ответа (Created: 201), возвращается объект со следующими *параметрами*. *Пример*.

Название	Схема JSON	Описание
id*	string	ID.
created_date*	date-time	Дата создания объекта.
modified_date*	date-time	Дата изменения объекта.
source_photo_name*	string	Имя файла, указанное для source_photo при создании объекта.
frame_coords_left	integer	Левая граница прямоугольника вокруг объекта.
frame_coords_top	integer	Верхняя граница прямоугольника вокруг объекта.
frame_coords_right	integer	Правая граница прямоугольника вокруг объекта.
frame_coords_bottom	integer	Нижняя граница прямоугольника вокруг объекта.
thumbnail*	uri	Миниатюра объекта.
active	boolean	true, если объект активен. Значение по умолчанию true.
features*	Поддерживается любой из форматов: strintfloatboolobjectarraynull	Признаки.
card*	integer	ID соответствующей карточки.

Пример ответа

```
{
  "card": 3,
  "created_date": "2023-01-16T08:12:55+00:00",
  "modified_date": "1970-01-01T00:00:00+00:00",
  "source_photo_name": "Смит_01.png",
  "source_photo": "http://172.23.218.94/uploads/cards/7w/3/face_%D0%A1%D0%BC%D0%B8%D1%82_01_ftgY5K.png",
  "thumbnail": "http://172.23.218.94/uploads/cards/TD/3/face_%D0%A1%D0%BC%D0%B8%D1%82_01_thumbnail_VyVsIj.png",
  "frame_coords_left": 757,
  "frame_coords_top": 79,
  "frame_coords_right": 948,
  "frame_coords_bottom": 353,
  "active": true,
```

(continues on next page)

(продолжение с предыдущей страницы)

```
"features": {},  
"id": "4493225067924944019",  
"meta": {}  
}
```

Полезные API запросы

```
GET /objects/bodies/  
POST /objects/bodies/  
GET /objects/bodies/{id}/  
DELETE /objects/bodies/{id}/  
PATCH /objects/bodies/{id}/  
GET /objects/cars/  
POST /objects/cars/  
GET /objects/cars/{id}/  
DELETE /objects/cars/{id}/  
PATCH /objects/cars/{id}/  
GET /objects/faces/  
POST /objects/faces/  
GET /objects/faces/{id}/  
DELETE /objects/faces/{id}/  
PATCH /objects/faces/{id}/
```

4.1.8 Видео

Вывести список видеоархив

Чтобы составить список видеоархивов, используйте следующий метод:

```
GET /videos/
```

Запрос содержит следующие *QUERY-STRING PARAMETERS*.

Название	Схема JSON	Описание
<code>case_in</code>	массив из integer	Выбрать видеоролики, относящиеся к нужным событиям.
<code>created_date_gt</code>	date-time	Выбрать объекты с <code>created_date</code> больше, чем указанное значение.
<code>created_date_gte</code>	date-time	Выбрать объекты со значением <code>created_date</code> больше или равным указанному.
<code>created_date_last_n_days</code>	integer	Выбрать объекты со значением <code>created_date</code> за последние N дней.
<code>created_date_lt</code>	date-time	Выбрать объекты со значением <code>created_date</code> меньше указанного.
<code>created_date_lte</code>	date-time	Выбрать объекты со значением <code>created_date</code> меньше или равно указанному.
<code>created_date_nth_full_week</code>	integer	Выбрать объекты со значением <code>created_date</code> за последние N недель (включая субботу и воскресенье)
<code>created_date_nth_work_week</code>	integer	Выбрать объекты со значением <code>created_date</code> за последние N недель (только рабочие дни, исключая субботу и воскресенье).
<code>limit</code>	string	Количество выводимых результатов.
<code>name*</code>	string	Выбрать видеоролики с указанным именем.
<code>ordering</code>	string	Допустимые значения: id, created_date, name.
<code>page</code>	string	Положение курсора.
<code>save_to</code>	string	Выбрать видеоархивы с указанным значением поля <code>save_to</code> .

Например, укажите произвольное значение для вывода результатов в поле `limit`.

CURL пример

```
curl -X GET "http://<findface-ip:port>/videos/?limit=2" \
-H "Accept: application/json" \
-H "Content-Language: ru" \
-H "Accept-Language: ru" \
-H "Authorization: Token <token>" \
```

В случае успешного ответа (ОК: 200) возвращается объект со следующими *параметрами*. *Пример*.

Название	Схема JSON
<code>next_page</code>	stringnull
<code>prev_page</code>	stringnull
<code>results</code>	массив

`results` содержат массив со следующими параметрами:

Название	Схема JSON	Описание
id*	integer	ID видеоархива.
camera_group*	integer	Группа камер.
name*	stringnull	Название видеоархива. Максимум 256 символа.
url	stringnull	URL.
camera	integernull	Камера.
processing_start_date*	date-time	Дата и время начала обработки видео.
active*	boolean	Обработка видео активна.
screenshot*	uri	URL скриншота.
stream_settings	{...}	Сериализатор <code>mixin</code> , который вызывает ошибку проверки <code>ValidationError</code> , если представлены избыточные поля. Может использоваться во вложенных сериализаторах.
source_len*	number	Длина источника в секундах.
health_status*	{...}	Дополнительная информация о статусе.
finished*	boolean	<code>true</code> , если обработка видео завершена.
queued*	boolean	<code>true</code> , если видео находится в очереди на обработку.
face_count*	integer	Количество созданных лиц.
file_size*	integer	Размер видеофайла в архиве.
created_date*	date-time	Дата создания объекта.
body_count*	integer	Количество созданных силуэтов.
car_count*	integer	Количество созданных автомобилей.
case	integernull	Происшествие.
face_cluster_count*	integernull	Количество созданных из видео кластеров лиц.
body_cluster_count*	integernull	Количество созданных из видео кластеров силуэтов.
car_cluster_count*	integernull	Количество созданных из видео кластеров автомобилей.

Примечание: Происшествия не поддерживаются в FindFace Multi 2.0 и хранятся только в API.

Пример ответа

```
{
  "next_page": null,
  "prev_page": null,
  "results": [
    {
      "id": 2,
      "camera_group": 1,
      "name": "file.mp4",
      "url": null,
      "camera": null,
      "processing_start_date": null,
      "active": false,
      "screenshot": "http://<findface-ip:port>/videos/2/screenshot/",
      "stream_settings": {
        "detectors": {
          "face": {
            "filter_max_size": 8192,
```

(continues on next page)

(продолжение с предыдущей страницы)

```

    "filter_min_quality": 0.45,
    "filter_min_size": 60,
    "fullframe_crop_rot": false,
    "fullframe_use_png": false,
    "jpeg_quality": 95,
    "overall_only": true,
    "post_best_track_frame": true,
    "post_best_track_normalize": true,
    "post_first_track_frame": false,
    "post_last_track_frame": false,
    "realtime_post_every_interval": false,
    "realtime_post_first_immediately": false,
    "realtime_post_interval": 1,
    "roi": "",
    "track_interpolate_bboxes": true,
    "track_max_duration_frames": 0,
    "track_miss_interval": 1,
    "track_overlap_threshold": 0.25,
    "track_send_history": false,
    "tracker_type": "simple_iou",
    "track_deep_sort_matching_threshold": 0.65,
    "track_deep_sort_filter_unconfirmed_tracks": true
  },
  "body": null,
  "car": null
},
"disable_drops": true,
"ffmpeg_format": "",
"ffmpeg_params": [],
"imotion_threshold": 0,
"play_speed": -1,
"rot": "",
"router_timeout_ms": 15000,
"router_verify_ssl": true,
"start_stream_timestamp": 0,
"stream_data_filter": "",
"use_stream_timestamp": false,
"video_transform": "",
"enable_recorder": false,
"enable_liveness": false
},
"source_len": null,
"health_status": {
  "enabled": false,
  "status": "DISABLED",
  "msg": "",
  "statistic": {
    "processed_duration": 0,
    "faces_posted": 0,
    "faces_failed": 0,
    "faces_not_posted": 0,
    "processing_fps": 0,

```

(continues on next page)

(продолжение с предыдущей страницы)

```

    "frames_dropped": 0,
    "frames_processed": 0,
    "frames_imotion_skipped": 0,
    "decoding_soft_errors": 0,
    "frame_width": 0,
    "frame_height": 0,
    "last_stream_timestamp": 0,
    "objects": null,
    "job_starts": 0
  },
  "code": "gray",
  "code_desc": "Обработка видео не запущена"
},
"finished": false,
"queued": false,
"face_count": 0,
"file_size": 2259950,
"created_date": "2023-01-12T08:57:36.811305Z",
"body_count": 0,
"car_count": 0,
"case": null,
"face_cluster_count": 0,
"body_cluster_count": 0,
"car_cluster_count": 0
}
]
}

```

Добавить новый видеоархив

Чтобы добавить новый видеоархив, используйте следующий метод:

```
POST /videos/
```

Тело запроса содержит application/json объект со следующими *параметрами*:

Название	Схема JSON	Описание
camera_group*	integer	Значение ID группы камер.
name*	stringnull	Название видеоархива (от 1 до 256 знаков).
url	stringnull	Минимум 1 символ.
camera	integernull	Камера.
stream_settings	{...}	Сериализатор mixin, который вызывает ошибку проверки ValidationError, если представлены избыточные поля. Может использоваться во вложенных сериализаторах.
case	integernull	Происшествие.

Примечание: Происшествия не поддерживаются в FindFace Multi 2.0 и хранятся только в API.

Пример запроса

Совет: Этот пример приведен только для ознакомления, подставьте свои значения в соответствующие поля. Вы можете заполнить только обязательные поля, а остальные будут по умолчанию.

```
{
  "camera_group": 0,
  "name": "A",
  "url": "A",
  "camera": 0,
  "stream_settings": {
    "detectors": {
      "face": {
        "filter_max_size": 0,
        "filter_min_quality": 0,
        "filter_min_size": 0,
        "fullframe_crop_rot": false,
        "fullframe_use_png": false,
        "jpeg_quality": 0,
        "overall_only": false,
        "post_best_track_frame": false,
        "post_best_track_normalize": false,
        "post_first_track_frame": false,
        "post_last_track_frame": false,
        "realtime_post_every_interval": false,
        "realtime_post_first_immediately": false,
        "realtime_post_interval": 0,
        "roi": "string",
        "track_interpolate_bboxes": false,
        "track_max_duration_frames": 0,
        "track_miss_interval": 0,
        "track_overlap_threshold": 0,
        "track_send_history": false,
        "tracker_type": "string",
        "track_deep_sort_matching_threshold": 0,
        "track_deep_sort_filter_unconfirmed_tracks": false
      },
      "body": {
        "filter_max_size": 0,
        "filter_min_quality": 0,
        "filter_min_size": 0,
        "fullframe_crop_rot": false,
        "fullframe_use_png": false,
        "jpeg_quality": 0,
        "overall_only": false,
        "post_best_track_frame": false,
        "post_best_track_normalize": false,
        "post_first_track_frame": false,
        "post_last_track_frame": false,
        "realtime_post_every_interval": false,
        "realtime_post_first_immediately": false,

```

(continues on next page)

(продолжение с предыдущей страницы)

```

    "realtime_post_interval": 0,
    "roi": "string",
    "track_interpolate_bboxes": false,
    "track_max_duration_frames": 0,
    "track_miss_interval": 0,
    "track_overlap_threshold": 0,
    "track_send_history": false,
    "tracker_type": "string",
    "track_deep_sort_matching_threshold": 0,
    "track_deep_sort_filter_unconfirmed_tracks": false
  },
  "car": {
    "filter_max_size": 0,
    "filter_min_quality": 0,
    "filter_min_size": 0,
    "fullframe_crop_rot": false,
    "fullframe_use_png": false,
    "jpeg_quality": 0,
    "overall_only": false,
    "post_best_track_frame": false,
    "post_best_track_normalize": false,
    "post_first_track_frame": false,
    "post_last_track_frame": false,
    "realtime_post_every_interval": false,
    "realtime_post_first_immediately": false,
    "realtime_post_interval": 0,
    "roi": "string",
    "track_interpolate_bboxes": false,
    "track_max_duration_frames": 0,
    "track_miss_interval": 0,
    "track_overlap_threshold": 0,
    "track_send_history": false,
    "tracker_type": "string",
    "track_deep_sort_matching_threshold": 0,
    "track_deep_sort_filter_unconfirmed_tracks": false
  }
},
"disable_drops": false,
"ffmpeg_format": "string",
"ffmpeg_params": [
  "A"
],
"imotion_threshold": 0,
"play_speed": 0,
"rot": "string",
"router_timeout_ms": 0,
"router_verify_ssl": false,
"start_stream_timestamp": 0,
"stream_data_filter": "string",
"use_stream_timestamp": false,
"video_transform": "string",
"enable_recorder": false,

```

(continues on next page)

(продолжение с предыдущей страницы)

```

    "enable_liveness": false
  },
  "case": 0
}

```

Вы можете отправить в запросе следующие параметры.

```

{
  "camera_group": 1,
  "name": "Pitt&Jolie"
}

```

CURL пример

```

curl -X POST "http://<findface-ip:port>/videos/" \
-H "Accept: application/json" \
-H "Content-Language: ru" \
-H "Accept-Language: ru" \
-H "Authorization: Token <token>" \
-H "Content-Type: application/json" \
-d '{"camera_group":1,"name":"Pitt&Jolie"}' \

```

Если ответ успешен (Created: 201), то возвращается объект, который содержит следующие *параметры*.
Пример.

Пример ответа

```

{
  "id": 6,
  "camera_group": 1,
  "name": "Pitt&Jolie",
  "url": null,
  "camera": null,
  "processing_start_date": null,
  "active": false,
  "screenshot": "http://<findface-ip:port>/videos/6/screenshot/",
  "stream_settings": {
    "detectors": {
      "face": {
        "filter_max_size": 8192,
        "filter_min_quality": 0.45,
        "filter_min_size": 60,
        "fullframe_crop_rot": false,
        "fullframe_use_png": false,
        "jpeg_quality": 95,
        "overall_only": true,
        "post_best_track_frame": true,
        "post_best_track_normalize": true,
        "post_first_track_frame": false,

```

(continues on next page)

(продолжение с предыдущей страницы)

```
"post_last_track_frame": false,
"realtime_post_every_interval": false,
"realtime_post_first_immediately": false,
"realtime_post_interval": 1,
"roi": "",
"track_interpolate_bboxes": true,
"track_max_duration_frames": 0,
"track_miss_interval": 1,
"track_overlap_threshold": 0.25,
"track_send_history": false,
"tracker_type": "simple_iou",
"track_deep_sort_matching_threshold": 0.65,
"track_deep_sort_filter_unconfirmed_tracks": true
},
"body": {
  "filter_max_size": 8192,
  "filter_min_quality": 0.6,
  "filter_min_size": 70,
  "fullframe_crop_rot": false,
  "fullframe_use_png": false,
  "jpeg_quality": 95,
  "overall_only": true,
  "post_best_track_frame": true,
  "post_best_track_normalize": true,
  "post_first_track_frame": false,
  "post_last_track_frame": false,
  "realtime_post_every_interval": false,
  "realtime_post_first_immediately": false,
  "realtime_post_interval": 1,
  "roi": "",
  "track_interpolate_bboxes": true,
  "track_max_duration_frames": 0,
  "track_miss_interval": 1,
  "track_overlap_threshold": 0.25,
  "track_send_history": false,
  "tracker_type": "simple_iou",
  "track_deep_sort_matching_threshold": 0.65,
  "track_deep_sort_filter_unconfirmed_tracks": true
},
"car": {
  "filter_max_size": 8192,
  "filter_min_quality": 0.73,
  "filter_min_size": 100,
  "fullframe_crop_rot": false,
  "fullframe_use_png": false,
  "jpeg_quality": 95,
  "overall_only": true,
  "post_best_track_frame": true,
  "post_best_track_normalize": true,
  "post_first_track_frame": false,
  "post_last_track_frame": false,
  "realtime_post_every_interval": false,
```

(continues on next page)

(продолжение с предыдущей страницы)

```

    "realtime_post_first_immediately": false,
    "realtime_post_interval": 1,
    "roi": "",
    "track_interpolate_bboxes": true,
    "track_max_duration_frames": 0,
    "track_miss_interval": 1,
    "track_overlap_threshold": 0.25,
    "track_send_history": false,
    "tracker_type": "simple_iou",
    "track_deep_sort_matching_threshold": 0.65,
    "track_deep_sort_filter_unconfirmed_tracks": true
  }
},
"source_len": null,
"health_status": {
  "enabled": false,
  "status": "WAITING_FOR_SYNC",
  "msg": "",
  "statistic": {},
  "code": "red",
  "code_desc": "Отсутствует задача в видео менеджере. Дождитесь синхронизации."
},
"finished": false,
"queued": false,
"face_count": 0,
"file_size": 0,
"created_date": "2023-01-18T08:38:52.119129Z",
"body_count": 0,
"car_count": 0,
"case": null,
"face_cluster_count": 0,
"body_cluster_count": 0,
"car_cluster_count": 0
}

```

Далее укажите `id`, чтобы загрузить файл с помощью запроса `PUT /videos/{id}/upload/source_file/`.

Загрузить видеофайл

Чтобы загрузить видеофайл, используйте следующий метод:

```
PUT /videos/{id}/upload/source_file/
```

Запрос содержит PATH PARAMETERS:

Название	Схема JSON	Описание
<code>id*</code>	<code>integer</code>	Уникальный идентификатор видео.

Прикрепите в тело запроса исходный видеофайл, указав `id` видеоархива.

Возвращает:

- Created: 201 – в случае успеха.
- Not Found: 404 – в случае ошибки.

CURL пример

```
curl -X PUT "http://<findface-ip:port>/videos/6/upload/source_file/" \  
-H "Content-Language: ru" \  
-H "Accept-Language: ru" \  
-H "Authorization: Token <token>" \  
-H "Content-Type: video/mp4" \  
--data-binary @pittjolie.mp4 \
```

Запуск обработки видеоархива

Чтобы запустить обработку видеоархива, используйте следующий метод:

```
POST /videos/{id}/process/
```

Запрос содержит id в PATH PARAMETERS, обозначающий уникальный идентификатор видеоархива.

CURL пример

```
curl -X POST "http://<findface-ip:port>/videos/6/process/" \  
-H "Content-Language: ru" \  
-H "Accept-Language: ru" \  
-H "Authorization: Token <token>" \
```

Возвращает:

- OK: 200 – в случае успеха.
- Not Found: 404 – в случае ошибки.

Полезные API запросы

```
GET /videos/  
POST /videos/  
GET /videos/{id}/  
PUT /videos/{id}/  
DELETE /videos/{id}/  
POST /videos/{id}/process/  
GET /videos/{id}/screenshot/  
POST /videos/{id}/screenshot/  
POST /videos/{id}/stop/  
PUT /videos/{id}/upload/source_file/  
GET /videos/count/
```

4.1.9 Группы камер

Группа камер - это набор камер, установленных в одном и том же месте или используемых для схожих целей.

Вывести список групп камер

Для того чтобы отобразить список групп камер, используйте следующий метод:

```
GET /camera-groups/
```

Запрос содержит следующие QUERY-STRING PARAMETERS:

Название	Схема JSON	Описание
limit	integer	Количество выводимых результатов.
ordering	string	Допустимые значения: id, created_date, modified_date.

Чтобы узнать id группы камеры, укажите значения для полей limit и ordering.

CURL пример

```
curl -X GET "http://<findface-ip:port>/camera-groups/?limit=3&ordering=id" \
  -H "Accept: application/json" \
  -H "Content-Language: ru" \
  -H "Accept-Language: ru" \
  -H "Authorization: Token <token>" \
```

В случае успешного ответа (ОК: 200) возвращается объект, содержащий массив результатов со следующими *параметрами*. *Пример*.

Совет: * – означает, что параметр является обязательным. – только для чтения.

Название	Схема JSON	Описание
id*	integer	ID группы камер.
created_date*	date-time	Дата создания объекта.
modified_date*	date-time	Дата изменения объекта.
active	boolean	Обрабатывать события из этой группы камер.
name*	string	Название группы камер, до 256 знаков.
comment	string	Комментарий, до 2048 символов.
deduplicate	boolean	True, если события из этой группы камер должны быть дедуплицированы.
deduplicateDelay	integer	Тайм-аут дедупликации событий (от 1 до 10000). Значение по умолчанию: 15.
labels	любое значение string	Метки, используемые для выделения определенного экземпляра <i>findface-video-worker</i> для обработки видеопотоков из этой группы камер.
permissions	любое значение string	Разрешения в данной группе камер.
face_threshold	numbernull	Порог схожести для лица (от 0 до 1).
body_threshold	numbernull	Порог схожести для силуэта (от 0 до 1).
car_threshold	numbernull	Порог схожести для автомобиля (от 0 до 1).

Пример ответа

```
{
  "results": [
    {
      "id": -1,
      "created_date": "2023-01-16T13:44:36.531501Z",
      "modified_date": "2023-01-16T13:44:36.531519Z",
      "active": true,
      "name": "Video archive default Camera Group",
      "comment": "",
      "deduplicate": false,
      "deduplicateDelay": 15,
      "labels": {},
      "permissions": {
        "1": "edit",
        "2": "view",
        "3": "view"
      },
      "face_threshold": null,
      "body_threshold": null,
      "car_threshold": null
    }
  ]
}
```

Создать группу камер

Чтобы добавить новую группу камер, используйте следующий метод:

```
POST /camera-groups/
```

Тело запроса обязательно и содержит application/json объект со следующими *параметрами*. *Пример*.

Название	Схема JSON	Описание
active	boolean	Обрабатывать события из этой группы камер.
name*	string	Название группы камер, до 256 знаков.
comment	string	Комментарий, до 2048 символов.
deduplicate	boolean	True, если события из этой группы камер должны быть дедуплицированы.
deduplicateDelay	integer	Тайм-аут дедупликации события (от 0 до 10000). Значение по умолчанию: 15.
labels	любое значение string	Метки, используемые для выделения определенного экземпляра <i>findface-video-worker</i> для обработки видеопотоков из этой группы камер.
permissions	любое значение string	Разрешения в данной группе камер.
face_threshold	numbernull	Порог схожести для лица (от 0 до 1).
body_threshold	numbernull	Порог схожести для силуэта (от 0 до 1).
car_threshold	numbernull	Порог схожести для автомобиля (от 0 до 1).

CURL пример

```
curl -X POST "http://<findface-ip:port>/camera-groups/" \
-H "Accept: application/json" \
-H "Content-Language: ru" \
-H "Accept-Language: ru" \
-H "Authorization: Token <token>" \
-H "Content-Type: application/json" \
-d '{"active":false,"name":"Test_camera_group","comment":"AAAAAA","deduplicate":false,
↪ "deduplicateDelay":0,"labels":{"},"permissions":{"},"face_threshold":0,"body_threshold
↪ ":0,"car_threshold":0}' \
```

Пример запроса

Совет: Этот пример приведен только для ознакомления, подставьте свои значения в соответствующие поля. Вы можете заполнить только обязательные поля, а остальные будут по умолчанию.

Вы можете отправить в запросе следующие параметры:

```
{
  "active": false,
```

(continues on next page)

(продолжение с предыдущей страницы)

```
"name": "Test_camera_group",
"comment": "AAAAAA",
"deduplicate": false,
"deduplicateDelay": 0,
"labels": {},
"permissions": {},
"face_threshold": 0,
"body_threshold": 0,
"car_threshold": 0
}
```

В случае успешного ответа (Created: 201) возвращается объект со следующими *параметрами*. *Пример*.

Пример ответа

```
{
  "id": 2,
  "created_date": "2023-01-23T12:14:09.885006Z",
  "modified_date": "2023-01-23T12:14:09.885030Z",
  "active": false,
  "name": "Test_camera_group",
  "comment": "AAAAAA",
  "deduplicate": false,
  "deduplicateDelay": 0,
  "labels": {},
  "permissions": {
    "1": "edit"
  },
  "face_threshold": 0,
  "body_threshold": 0,
  "car_threshold": 0
}
```

Используйте id группы камер для того, чтобы отправить POST запрос для *создания новой камеры*.

Полезные API запросы

```
GET /camera-groups/
POST /camera-groups/
GET /camera-groups/{id}/
DELETE /camera-groups/{id}/
PATCH /camera-groups/{id}/
GET /camera-groups/count/
```

4.1.10 Камеры

Создать новую камеру

Чтобы создать новую камеру, используйте следующий метод:

```
POST /cameras/
```

Тело запроса содержит application/json объект со следующими обязательными *параметрами*:

name*	Схема JSON	Описание
group*	integer	Группа камер.
name*	string	Название камеры (от 1 до 256 знаков).

Остальные параметры см. <http://<findface-ip:port>/api-docs>.

Вы можете отправить в запросе следующие параметры, подставив свои значения:

Пример запроса

```
{
  "group": 1,
  "active": false,
  "name": "Camera_name",
  "comment": "AAAAAA",
  "url": "url"
}
```

CURL пример

```
curl -X POST "http://<findface-ip:port>/cameras/" \
-H "Accept: application/json" \
-H "Content-Language: ru" \
-H "Accept-Language: ru" \
-H "Authorization: Token <token>" \
-H "Content-Type: application/json" \
-d '{"group":1,"active":false,"name":"Camera_name","comment":"AAAAAA","url":"url"}' \
```

Возвращает

- Created: 201 – в случае успеха.

4.1.11 Поиск лиц в системе

FindFace Multi позволяет выполнять поиск лиц по всей системе.

Вы можете задать поиск лиц одним из следующих способов:

- загрузив фотографию
- по ID события
- по ID объекта
- по ID кластера

Чтобы найти человека по загруженной фотографии, выполните следующие действия:

1. Загрузите фотографию с помощью *метода* `POST /detect`.
2. Прикрепите исходную фотографию и отправьте `POST` запрос.
3. В ответе вы получите `id` объекта и координаты границ лица.
4. Если на изображении несколько лиц, выберите одно из них, нужное вам. Скопируйте `id` нужного вам объекта и используйте его для поиска с помощью *метода* `GET /cards/humans/` в поле `looks_like` следующим образом: `detection:<detection id>`.
5. Вы также можете использовать:
 - `event:<event id>` (для лица, силуэта или автомобиля)
 - `object:<object id>` (для лица, силуэта или автомобиля)
 - `cluster:<cluster id>`
6. Вы увидите результаты поиска. *Пример*. При необходимости можно сузить область поиска, указав список наблюдения, порог сходства и т. д.

4.1.12 Сравнение двух лиц

FindFace Multi позволяет сравнить два лица и убедиться в их принадлежности одному человеку.

Вы можете сравнить два объекта из разных источников или объект с объектом внутри одной и той же карточки и вернуть сходство между ними.

Выполните следующие действия:

1. Выполните *обнаружение лица (автомобиля, силуэта)* на прикрепленном изображении и скопируйте `id` интересующего вас объекта.
2. Используйте метод `GET /verify`.

`GET /verify`

Запрос содержит следующие QUERY-STRING PARAMETERS:

Название	Схема JSON	Описание
card_id	string	Минимум 1 символ. В этом поле указывают ID карты, в которой хранится объект для сравнения.
*object1	string	Минимум 1 символ. Это поле может содержать одну из следующих ссылок: <ul style="list-style-type: none"> • detection:<id> – для обнаруженного объекта • faceevent:<event id> – для события «лицо» • carevent:<event id> – для события «автомобиль» • bodyevent:<event id> – для события «силуэт» • faceobject:<face id> – для объекта «лицо» • carobject:<face id> – для объекта «автомобиль» • carobject:<face id> – для объекта «силуэт»
object2	string	Минимум 1 символ. Аналогично как для object1.

CURL пример

```
curl -X GET "http://<findface-ip:port>/verify?card_id=1&object1=detection
↪%3Acf58e4uv54rotim9jtd0&object2=detection%3Acf58g1mv54rotim9jtdg" \
-H "Accept: application/json" \
-H "Content-Language: ru" \
-H "Accept-Language: ru" \
-H "Authorization: Token <token>" \
```

В случае успешного ответа (ОК: 200), возвращается объект, содержащий уровень схожести.

Название	Схема JSON	Описание
confidence*	number	Оценка сходства лиц (от 0 до 1).

Пример ответа

```
{
  "confidence": {
    "face_objects": {
      "4493493039043981648": 0.7896046
    },
    "average_conf": 0.7896046
  }
}
```

4.2 Вебхуки

Вы можете настроить FindFace Multi для автоматической отправки уведомлений об определенных событиях, эпизодах, скриншотах счетчиков на заданный URL-адрес. Для этого создайте и настройте вебхук. При возникновении нужного события/эпизода/скриншота счетчика FindFace Multi отправит HTTP-запрос на URL-адрес, указанный в настройках вебхука.

Вебхуки можно использовать для решения разнообразных задач, например, для уведомления пользователя об определенном событии, вызова определенных действий на целевом веб-сайте, при решении задач безопасности, таких как удаленное автоматическое управление доступом и др.

В этом разделе:

- *Настройка вебхука*
- *Как работает вебхук*
- *Подробные данные в вебхуках (Verbose)*

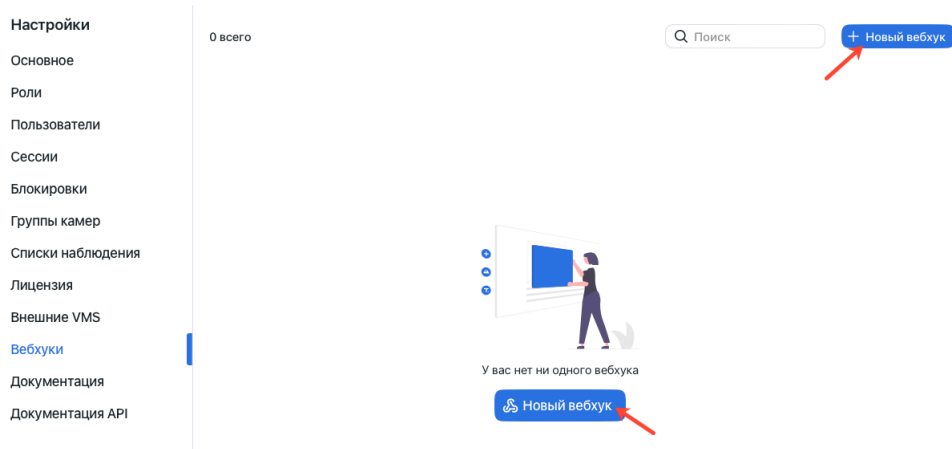
4.2.1 Настройка вебхука

Важно: Для создания вебхука необходимы права администратора.

Примечание: Перед использованием вебхуков убедитесь, что хотя бы один из следующих параметров задан в файле `/opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py`: `SERVICE_EXTERNAL_ADDRESS` или `EXTERNAL_ADDRESS`.

Для создания вебхука выполните следующие действия:

1. Перейдите на вкладку *Настройки*. Выберите *Вебхуки*.
2. Нажмите *+ Новый вебхук*.



3. Введите заголовок вебхука.

Контроль доступа

Изменено X

X

↗

Заголовок вебхука

Контроль доступа

URL

Количество сообщений в пакете

1

Количество попыток отправки

10

0 - не ограничено

Фильтры

```
{
  "face_events": {
    "matched_lists_in": [],
    "camera_group_in": [],
    "camera_in": [],
    "matched_card_in": [],
    "matched": true,
    "confidence_gte": 0.75,
    "allowed_bs_types": [
      "overall",
      "realtime"
    ],
    "temperature_gte": 37,
    "liveness_gte": 0.65,
    "headpose_yaw_angle_gte": -180,
    "headpose_yaw_angle_lte": 180,
  }
}
```

Сохранить

Отменить

4. Укажите адрес, на который будут отправляться оповещения.
5. Вы можете отправлять уведомления в пакетном режиме. Укажите максимальное количество уведомлений в пакете. Фактическое количество может быть меньшим.
6. Укажите максимальное количество попыток отправить уведомление. Интервал между попытками увеличивается экспоненциально с максимумом 100 секунд.

Важно: Для того чтобы получить все сообщения с момента потери соединения, если оно имеет место, установите значение 0. Установите 1, чтобы пропустить старые сообщения.

7. FindFace Multi будет автоматически отправлять оповещения о событиях, эпизодах, скриншотах счетчика, удовлетворяющих заданным фильтрам. Фильтровать события можно по следующим параметрам:

Универсальные фильтры для событий распознавания (лицо, силуэт, транспортное средство):

Секции: `face_events`, `body_events`, `car_events`.

- `"allowed_bs_types"`: режим отслеживания объектов на видео, возможные значения: `overall`, `realtime`.
- `"camera_group_in"`: id группы камер, число.
- `"camera_in"`: id камеры, число.
- `"confidence_gte"`: минимальное значение уверенности алгоритма, число.
- `"matched_lists_in"`: id списка наблюдения, число.
- `"matched_card_in"`: id совпавшей карточки, число.
- `"matched"`: результат сопоставления с картотекой, логическое значение. Установите `true` или `false`, если отправку оповещения должны вызывать только события с совпадениями или только события без совпадений.

Фильтры для событий распознавания лиц:

Секция: `face_events`.

- `"liveness_gte"`: минимальное значение уверенности алгоритма в том, что лицо витально, десятичное число.
- `"headpose_yaw_angle_gte"`: минимальный поворот головы, число (градусы). Если голова полностью повернута влево, угол равен `-180`. Если полностью повернута вправо, то `180`. Нейтральное положение головы соответствует `0`.
- `"headpose_yaw_angle_lte"`: максимальный поворот головы, число (градусы). Если голова полностью повернута влево, угол равен `-180`. Если полностью повернута вправо, то `180`. Нейтральное положение головы соответствует `0`.
- `"headpose_pitch_angle_gte"`: минимальный наклон головы, число (градусы). Если голова наклонена вперед, а подбородок касается груди, угол равен `-180`. Если голова наклонена назад и обращена к небу, то `180`. Нейтральное положение головы соответствует `0`.
- `"headpose_pitch_angle_lte"`: максимальный наклон головы, число (градусы). Если голова наклонена вперед, а подбородок касается груди, угол равен `-180`. Если голова наклонена назад и обращена к небу, то `180`. Нейтральное положение головы соответствует `0`.
- `"gender_in"`: пол, словарь строк. Возможные значения: `"male"`, `"female"`.
- `"age_lte"`: максимальный возраст, число.
- `"age_lte"`: минимальный возраст, число.
- `"glasses_in"`: очки, словарь строк. Возможные значения: `"none"`, `"eye"` (очки), `"sun"` (солнцезащитные очки).
- `"emotions_in"`: эмоции, словарь строк. Возможные значения: `"any"`, `"angry"`, `"fear"`, `"sad"`, `"neutral"`, `"disgust"`, `"happy"`, `"surprise"`.
- `"beard_in"`: борода, словарь строк. Возможные значения: `"none"`, `"beard"`.
- `"medmask_in"`: медицинская маска, словарь строк. Возможные значения: `"correct"`, `"incorrect"`, `"none"`.
- `"liveness_in"`: оценка витальности `liveness`, словарь строк. Возможные значения: `"real"`, `"fake"`.

Фильтры для событий распознавания силуэтов:

Секция: `body_events`.

- `"headwear_in"`: тип головного убора (шапка/кепка, капюшон/платок, без убора), словарь строк. Возможные значения: `"hat"`, `"hood"`, `"none"`.
- `"upper_clothes_in"`: обобщенная категория одежды верхней части тела (с длинными рукавами, с короткими рукавами, без рукавов), словарь строк. Возможные значения: `"long_sleeves"`, `"short_sleeves"`, `"without_sleeves"`.
- `"detailed_upper_clothes_in"`: детализированный тип одежды верхней части тела (куртка, пальто, жилет без рукавов, толстовка, футболка, рубашка, платье), словарь строк. Возможные значения: `"jacket"`, `"coat"`, `"sleeveless"`, `"sweatshirt"`, `"t-shirt"`, `"shirt"`, `"dress"`.
- `"top_color_in"`: цвет одежды верхней части тела, словарь строк. Возможные значения: `white`, `"black"`, `"grey"`, `"brown"`, `"red"`, `"orange"`, `"yellow"`, `"green"`, `"lightblue"`, `"blue"`, `"purple"`, `"pink"`, `"beige"`, `"violet"`.
- `"lower_clothes_in"`: тип одежды нижней части тела (брюки, юбка, шорты, неопределенный), словарь строк. Возможные значения: `"pants"`, `"obsured"`, `"skirt"`, `"shorts"`.
- `"bottom_color_in"`: цвет одежды нижней части тела, словарь строк. Возможные значения: `white`, `"black"`, `"grey"`, `"brown"`, `"red"`, `"orange"`, `"yellow"`, `"green"`, `"lightblue"`, `"blue"`, `"purple"`, `"pink"`, `"beige"`, `"violet"`.
- `"bag_hand_in"`: наличие сумки в руке(ах), словарь строк. Возможные значения: `"none"`, `"hand"`.
- `"bag_back_in"`: наличие сумки на спине, словарь строк. Возможные значения: `"none"`, `"back"`.
- `"vest_type_score_gte"`: минимальная достоверность распознавания наличия жилета, число.
- `"vest_type_score_lte"`: максимальная достоверность распознавания наличия жилета, число.
- `"vest_type_in"`: наличие средства индивидуальной защиты в виде жилета, словарь строк. Возможные значения: `"green/yellow"`, `"orange"`, `"not_visible"`, `"none"`.
- `"helmet_type_score_lte"`: максимальная достоверность распознавания наличия шлема, число.
- `"helmet_type_score_gte"`: минимальная достоверность распознавания наличия шлема, число.
- `"helmet_type_in"`: наличие средства индивидуальной защиты в виде шлема, словарь строк. Возможные значения: `"red/orange"`, `"white"`, `"other"`, `"not_visible"`, `"none"`.
- `"age_group_score_lte"`: максимальная достоверность распознавания возрастной группы, число.
- `"age_group_score_gte"`: минимальная достоверность распознавания возрастной группы, число.
- `"age_group_in"`: возраст по группе, словарь строк. Возможные значения: `"0-16"`, `"17-35"`, `"36-50"`, `"50+"`.
- `"gender_score_lte"`: максимальная достоверность распознавания пола, число.
- `"gender_score_gte"`: минимальная достоверность распознавания пола, число.

- "gender_in": пол, словарь строк. Возможные значения: "male", "female".

Фильтры для событий распознавания транспортных средств:

Секция: car_events.

- "category_confidence_gte": минимальная достоверность распознавания категории транспортного средства, число.
- "category_confidence_lte": максимальная достоверность распознавания категории транспортного средства, число.
- "category_type_in": категория транспортного средства, словарь строк. Возможные значения: "unknown", "A", "B", "BE", "C", "CE", "D", "DE", "other".
- "color_in" цвет транспортного средства, словарь строк. Возможные значения: "beige", "black", "blue", "brown", "cyan", "gold", "green", "grey", "orange", "pink", "purple", "red", "violet", "white", "yellow", "coming_soon", "unknown".
- "body_in": тип кузова транспортного средства, словарь строк. Возможные значения: "suv", "sedan", "crossover", "convertible", "coupe", "wagon", "cab", "minibus", "minivan", "limousine", "coming_soon", "unknown".
- "make_in": марка транспортного средства, словарь строк. Ознакомьтесь с соответствующим фильтром событий, чтобы узнать, какие производители поддерживаются в текущей версии.
- "model_in": модель транспортного средства, словарь строк. Ознакомьтесь с соответствующим фильтром событий, чтобы узнать, какие модели поддерживаются в текущей версии.
- "license_plate_number_in": регистрационный номер транспортного средства, словарь строк. Подстановочные знаки не поддерживаются.
- "license_plate_color_in": цвет номерного знака, словарь строк. Возможные значения: "unknown", "white", "yellow", "blue", "green", "grey".
- "license_plate_color_confidence_lte": максимальная достоверность распознавания цвета номерного знака, число.
- "license_plate_color_confidence_gte": минимальная достоверность распознавания цвета номерного знака, число.
- "license_plate_country_in": страна номерного знака, словарь строк. Ознакомьтесь с соответствующим фильтром событий, чтобы узнать, какие страны поддерживаются в текущей версии.
- "license_plate_region_in": регион номерного знака, словарь строк. Ознакомьтесь с соответствующим фильтром событий, чтобы узнать, какие регионы поддерживаются в текущей версии.
- "special_vehicle_type_in": спецназначение транспортного средства, словарь строк. Возможные значения: "taxi", "route_transport", "car_sharing", "ambulance", "police", "rescue_service", "gas_service", "military", "road_service", "other_special", "not_special", "unknown".
- "weight_type_in": весовая категория транспортного средства. Возможные значения: "B_light", "B_heavy", "C_light", "C_heavy", "D_light", "D_long", "other".
- "weight_type_confidence_lte": максимальная достоверность распознавания весовой категории транспортного средства, число.
- "weight_type_confidence_gte": минимальная достоверность распознавания весовой категории транспортного средства, число.

- "orientation_in": ракурс транспортного средства, словарь строк. Возможные значения: "front", "back", "side".
- "orientation_confidence_lte": максимальная достоверность распознавания ракурса автомобиля, число.
- "orientation_confidence_gte": минимальная достоверность распознавания ракурса автомобиля, число.

Эпизоды с людьми:

Секция: human_episodes.

- "allowed_types": статус эпизода, возможные значения: открытие эпизода (episode_open), добавление нового события в эпизод (episode_event), закрытие эпизода (episode_close).
- "camera_group_in": id группы камер, число.
- "camera_in": id камеры, число.
- "matched_lists_in": id списка наблюдения, число.
- "face_matched": статус совпадения лица в эпизоде с картотекой, логическое значение. Установите true, если только эпизоды с совпадениями лиц, или false, если только эпизоды без совпадений лиц должны вызывать отправку оповещения.
- "body_matched": статус совпадения силуэта в эпизоде с картотекой, логическое значение. Установите true, если только эпизоды с совпадениями силуэтов, или false, если только эпизоды без совпадений силуэтов должны вызывать отправку оповещения.
- "events_count_gte": минимальное количество событий в эпизоде, число.
- "events_count_lte": максимальное количество событий в эпизоде, число.

Эпизоды с транспортными средствами:

Секция: car_episodes.

- "allowed_types": статус эпизода, возможные значения: открытие эпизода (episode_open), добавление нового события в эпизод (episode_event), закрытие эпизода (episode_close).
- "camera_group_in": id группы камер, число.
- "camera_in": id камеры, число.
- "matched_lists_in": id списка наблюдения, число.
- "car_matched": статус совпадения транспортного средства в эпизоде с картотекой, логическое значение. Установите true, если только эпизоды с совпадениями ТС, или false, если только эпизоды без совпадений ТС должны вызывать отправку оповещения.
- "events_count_gte": минимальное количество событий в эпизоде, число.
- "events_count_lte": максимальное количество событий в эпизоде, число.

Скриншоты счетчиков:

Секция: counters.

- "counter_in": id счетчика, число.
- "camera_group_in": id группы камер, число.
- "camera_in": id камеры, число.
- "faces_gte": минимальное количество лиц в скриншоте счетчика, число.
- "faces_lte": максимальное количество лиц в скриншоте счетчика, число.
- "silhouettes_gte": минимальное количество силуэтов в скриншоте счетчика, число.
- "silhouettes_lte": максимальное количество силуэтов в скриншоте счетчика, число.
- "cars_gte": минимальное количество транспортных средств в скриншоте счетчика, число.
- "cars_lte": максимальное количество транспортных средств в скриншоте счетчика, число.
- "proximity_min_lte": отправить оповещение, если минимальное задетектированное расстояние в метрах меньше указанной величины, число.
- "proximity_min_gte": отправить оповещение, если минимальное задетектированное расстояние в метрах больше указанной величины, число.
- "proximity_avg_lte": отправить оповещение, если среднее задетектированное расстояние в метрах меньше указанной величины, число.
- "proximity_avg_gte": отправить оповещение, если среднее задетектированное расстояние в метрах больше указанной величины, число.
- "proximity_max_lte": отправить оповещение, если максимальное задетектированное расстояние в метрах меньше указанной величины, число.
- "proximity_max_gte": отправить оповещение, если максимальное задетектированное расстояние в метрах больше указанной величины, число.

```
{
  "face_events": {
    "matched_lists_in": [],
    "camera_group_in": [],
    "camera_in": [],
    "matched_card_in": [],
    "matched": true,
    "confidence_gte": 0.75,
    "allowed_bs_types": [
      "overall",
      "realtime"
    ],
    "liveness_gte": 0.65,
    "headpose_yaw_angle_gte": -180,
    "headpose_yaw_angle_lte": 180,
    "headpose_pitch_angle_gte": -180,
    "headpose_pitch_angle_lte": 180,
    "gender_in": [],
    "age_lte": 0,
    "age_gte": 0,
    "glasses_in": [],
```

(continues on next page)

(продолжение с предыдущей страницы)

```

"emotions_in": [],
"beard_in": [],
"medmask_in": [],
"liveness_in": []
},
"body_events": {
"matched_lists_in": [],
"camera_group_in": [],
"camera_in": [],
"matched_card_in": [],
"matched": true,
"confidence_gte": 0.75,
"allowed_bs_types": [
"overall",
"realtime"
],
"headwear_in": [],
"upper_clothes_in": [],
"detailed_upper_clothes_in": [],
"top_color_in": [],
"lower_clothes_in": [],
"bottom_color_in": [],
"bag_hand_in": [],
"bag_back_in": [],
"vest_type_score_gte": 0,
"vest_type_score_lte": 0,
"vest_type_in": [],
"helmet_type_score_lte": 0,
"helmet_type_score_gte": 0,
"helmet_type_in": [],
"age_group_score_lte": 0,
"age_group_score_gte": 0,
"age_group_in": [],
"gender_score_lte": 0,
"gender_score_gte": 0,
"gender_in": []
},
"car_events": {
"matched_lists_in": [],
"camera_group_in": [],
"camera_in": [],
"matched_card_in": [],
"matched": true,
"confidence_gte": 0.75,
"allowed_bs_types": [
"overall",
"realtime"
],
"category_confidence_gte": 0,
"category_confidence_lte": 0,
"category_type_in": [],
"color_in": [],

```

(continues on next page)

(продолжение с предыдущей страницы)

```
"body_in": [],
"make_in": [],
"model_in": [],
"license_plate_number_in": [],
"license_plate_color_in": [],
"license_plate_color_confidence_lte": 0,
"license_plate_color_confidence_gte": 0,
"license_plate_country_in": [],
"license_plate_region_in": [],
"special_vehicle_type_in": [],
"weight_type_in": [],
"weight_type_confidence_lte": 0,
"weight_type_confidence_gte": 0,
"orientation_in": [],
"orientation_confidence_lte": 0,
"orientation_confidence_gte": 0
},
"human_episodes": {
  "allowed_types": [
    "episode_open",
    "episode_event",
    "episode_close"
  ],
  "matched_lists_in": [],
  "camera_in": [],
  "camera_group_in": [],
  "events_count_gte": 0,
  "events_count_lte": 999,
  "face_matched": true,
  "body_matched": true },
"car_episodes": {
  "allowed_types": [
    "episode_open",
    "episode_event",
    "episode_close"
  ],
  "matched_lists_in": [],
  "camera_in": [],
  "camera_group_in": [],
  "events_count_gte": 0,
  "events_count_lte": 999,
  "car_matched": true },
"counters": {
  "counter_in": [],
  "camera_in": [],
  "camera_group_in": [],
  "faces_gte": 1,
  "faces_lte": 100,
  "silhouettes_gte": 1,
  "silhouettes_lte": 100,
  "cars_gte": 1,
  "cars_lte": 100,
```

(continues on next page)

(продолжение с предыдущей страницы)

```

"proximity_min_lte": 100,
"proximity_min_gte": 0,
"proximity_avg_lte": 100,
"proximity_avg_gte": 0,
"proximity_max_lte": 100,
"proximity_max_gte": 0
}
}

```

Важно: Используйте только фильтры, соответствующие цели поиска. Для выключения фильтра удалите его из вебхука. Не оставляйте фильтр пустым ({}), поскольку в этом случае фильтр вернет пустой результат.

Примечание: Для получения всех оповещений передайте скобки без вложенных фильтров:

```
{}
```

Совет: Пример №1. Получение оповещений обо всех событиях с транспортными средствами:

```
{ "car_events": {} }
```

Пример №2. Получение оповещений об открытии эпизодов с людьми с совпадениями в картотеке по лицам и силуэтам:

```

{ "human_episodes": { "allowed_types": ["episode_open"], "face_matched": true,
  ↳ "body_matched": true }}

```

Примечание: Вы можете задать несколько значений в фильтрах с квадратными скобками. В этом случае вебхук будет активироваться при совпадении одного из значений фильтра. В примере ниже вы будете оповещены о событии с силуэтом с группы камер 1 или 3, если совпала карточка с id 12 или 25.

```

{
  "body_events": {
    "camera_group_in": [1, 3],
    "matched_card_in": [12, 25],
  },
}

```

8. Поставьте флажок *Активная*.
9. Нажмите *Сохранить*.

4.2.2 Как работает вебхук

Для тестирования работы вебхука можно использовать следующий простой веб-сервер на Python, обеспечивающий захват отправленных вебхуком оповещений о событиях:

```
from pprint import pprint
from aiohttp import web

async def handle(request):
    pprint(await request.json())
    return web.Response(status=200)

app = web.Application()
# for aiohttp v 3.x
# app.add_routes([web.post('/', handle)])

# for aiohttp v 2.x
app.router.add_post('/', handle)

web.run_app(app, port=8888)
```

Важно: Используемый вами сервер-ловушка оповещений по вебхуку должен возвращать ответ HTTP 200 после получения вебхука от FindFace Multi, по аналогии с примером выше.

Если для вебхука не задано ни одного фильтра, данный веб-сервер будет получать оповещения обо всех произошедших в системе событиях, эпизодах, скриншотах счетчиков.

Для просмотра статуса отправки вебхука в FindFace Multi выполните следующую команду:

```
sudo journalctl CONTAINER_NAME=findface-multi-findface-multi-legacy-1 -f | grep 'Webhook'
```

4.2.3 Подробные данные в вебхуках (Verbose)

По умолчанию оповещения по вебхуку содержат только идентификаторы таких сущностей, как карточки, списки наблюдения, камеры и группы камер. Для того чтобы получать подробную информацию по данным сущностям, включите подробный режим оповещений по вебхукам (режим verbose).

Для этого откройте файл конфигурации `/opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py` и установите в нем `'VERBOSE_WEBHOOKS': True`:

```
sudo vi /opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py

...
FFSECURITY = {
    ...
    # send serialized cards, card-lists, camera and camera groups in webhooks
    'VERBOSE_WEBHOOKS': True,
    ...
}
...
```

После внесения изменений обязательно перезапустите контейнер `findface-multi-findface-multi-legacy-1`.

```
sudo docker container restart findface-multi-findface-multi-legacy-1
```

4.3 Внешние системы управления видео (VMS)

На стороне FindFace Multi реализован механизм интеграции с внешними системами управления видео (VMS), позволяющий добавлять камеры из них в FindFace Multi и осуществлять их обработку. После добавления камеры из VMS для нее становятся доступны все действия, которые доступны для обычной камеры FindFace Multi.

Интеграция с внешними VMS возможна через плагин. Взаимодействие между плагином и FindFace Multi осуществляется по HTTP. Плагин взаимодействует с VMS, FindFace Multi – с плагином.

В этой главе:

- *Создание сервисной учетной записи для плагина интеграции с VMS*
- *FindFace Multi*
- *PostgreSQL*
- *Установка и настройка плагина для интеграции VMS с FindFace*
- *Добавление Внешних VMS в FindFace Multi*

Этот раздел содержит пошаговое руководство по установке и настройке функции внешних VMS в FindFace Multi. Пользователь должен иметь большой опыт администрирования ОС Linux и контейнеров Docker.

4.3.1 Создание сервисной учетной записи для плагина интеграции с VMS

Создайте пользователя (имя пользователя: **ntech**) с определенным набором разрешений, который будет использоваться в качестве сервисной учетной записи для плагина интеграции. Для этого выполните приведенную ниже команду:

```
adduser --system --disabled-password --disabled-login --home /var/empty \
        --no-create-home --quiet --force-badname --group ntech
```

4.3.2 FindFace Multi

1. Откройте файл `/opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py` и найдите раздел `External VMS Integration settings`.
2. Измените параметр `ENABLED` на `True`.
3. Запомните значение `TOKEN`, так как оно вам понадобится для настройки плагина позже. (Значение по умолчанию – `VmsPluginToken PLUGIN_TOKEN`).

```
...
# -- External VMS integration settings --
# cleanup settings
'EXTERNAL_VMS_EVENTS_MAX_AGE': 0,
'EXTERNAL_VMS_SEND_EVENTS_STATUS_MAX_AGE': 0,
'EXTERNAL_VMS': {
  'ENABLED': True,
  'PLUGIN_ADDRESS': 'http://127.0.0.1:18333',
  'TOKEN': 'VmsPluginToken PLUGIN_TOKEN',
  'EVENT_SENDER': {
    'ENABLED': True,
    'ALLOWED_TYPES': ['face'],
    'SENDER_TASKS': 1,
    'MAX_SEND_ATTEMPTS': 1,
    'MIN_EVENT_SEND_TIMEOUT': 0.1,
    'MAX_EVENT_SEND_TIMEOUT': 100,
    'RESPONSE_TIMEOUT': 10,
  },
},
...
```

Совет: В нашем примере мы устанавливаем плагин на тот же хост, где развернут FindFace Multi. В случае использования отдельного компьютера, пожалуйста, укажите правильный адрес сервера, на котором находится плагин интеграции с VMS, в поле `PLUGIN_ADDRESS`.

Примечание: Если вам нужно отправлять события на VMS, пожалуйста, также установите значение `ENABLED`, равным `True` в разделе `EVENT_SENDER`.

4. Перезапустите контейнер `findface-multi-findface-multi-legacy-1`.

```
sudo docker container restart findface-multi-findface-multi-legacy-1
```

Краткое описание параметров приведено ниже:

Параметр	Описание
EXTERNAL_VMS_EVENTS_MAX_AGE	Время хранения событий, полученных от VMS (дни). 0 – неограниченно.
EXTERNAL_VMS_SEND_EVENTS_MAX_AGE	Время хранения событий, отправленных в VMS (дни). 0 – неограниченно.
EXTERNAL_VMS_ENABLED	Статус модуля интеграции VMS с FindFace. Если ENABLED , то в пользовательском интерфейсе появится опция “Внешние VMS”.
PLUGIN_ADDRESS	IP адрес плагина.
TOKEN	Токен, необходимый плагину интеграции с VMS для аутентификации в Findface Multi (должен соответствовать токену в файле конфигурации плагина).
EVENT_SENDER_ENABLED	Если True , все совпадающие события в Findface Multi будут отправлены во внешние VMS.
ALLOWED_TYPES	Типы отправляемых объектов. В настоящее время поддерживаются только объекты «Лица».
SENDER_TASKS	Количество одновременных задач для отправки событий в плагин. Значение по умолчанию = 1. При необходимости оно может быть увеличено. За дополнительной информацией обратитесь к нашим экспертам за консультацией (support@ntechlab.com).
MAX_SEND_ATTEMPTS	Количество попыток отправки до того, как она будет признана неудачной.
MIN_EVENT_SEND_INTERVAL	Время между попытками отправки, если произошла неудачная отправка.
MAX_EVENT_SEND_INTERVAL	Порог экспоненциально от минимума к максимуму до тех пор, пока увеличивается количество попыток MAX_SEND_ATTEMPTS.
RESPONSE_TIMEOUT	Время ожидания ответа от плагина при отправке события.

4.3.3 PostgreSQL

1. Откройте файл `/opt/findface-multi/docker-compose.yaml` и запишите значение `POSTGRES_PASSWORD`. Оно понадобится для следующего шага.

```
sudo vi /opt/findface-multi/docker-compose.yaml
...
postgresql:
  environment: {POSTGRESQL_ALLOW_REMOTE_CONNECTIONS: 'no', POSTGRES_
  ↪PASSWORD: POSTGRES_PASSWORD}
...
```

2. Войдите в контейнер `findface-multi-postgresql-1`, выполнив следующую команду:

```
sudo docker exec -i findface-multi-postgresql-1 /bin/bash -c "PGPASSWORD=
  ↪{POSTGRES_PASSWORD} psql --username postgres"
```

Чтобы убедиться, что вы успешно вошли в контейнер, используйте команду `\l` для просмотра списка баз данных:

```
\l
      Name | Owner | List of databases
Encoding | Collate | Ctype | Access privileges
-----
ffcounter | ntech | UTF8 | C.UTF-8 | C.UTF-8 |
ffsecurity | ntech | UTF8 | C.UTF-8 | C.UTF-8 |
ffsecurity_audit | ntech | UTF8 | C.UTF-8 | C.UTF-8 |
ffsecurity_identity_provider | ntech | UTF8 | C.UTF-8 | C.UTF-8 |
postgres | postgres | UTF8 | en_US.UTF-8 | en_US.UTF-8 |
template0 | postgres | UTF8 | en_US.UTF-8 | en_US.UTF-8 | =c/postgres, postgres=CTc/postgres,
template1 | postgres | UTF8 | en_US.UTF-8 | en_US.UTF-8 | =c/postgres, postgres=CTc/postgres
(7 rows)
```

3. Выполните следующую команду, чтобы создать новую базу данных, необходимую для работы VMS:

```
CREATE DATABASE ffsintegration WITH OWNER ntech ;
```

4. Подтвердите создание базы данных, снова выполнив команду \1. После завершения используйте параметр \q для выхода из PostgreSQL.

```
CREATE DATABASE ffsintegration WITH OWNER ntech ;
CREATE DATABASE
\1
```

Name	Owner	Encoding	Collate	Ctype	Access privileges
ffcounter	ntech	UTF8	C.UTF-8	C.UTF-8	
ffsecurity	ntech	UTF8	C.UTF-8	C.UTF-8	
ffsecurity_audit	ntech	UTF8	C.UTF-8	C.UTF-8	
ffsecurity_identity_provider	ntech	UTF8	C.UTF-8	C.UTF-8	
ffsintegration	ntech	UTF8	en_US.UTF-8	en_US.UTF-8	
postgres	postgres	UTF8	en_US.UTF-8	en_US.UTF-8	=c/postgres +
template0	postgres	UTF8	en_US.UTF-8	en_US.UTF-8	postgres=Ctc/postgres +
template1	postgres	UTF8	en_US.UTF-8	en_US.UTF-8	=c/postgres +
					postgres=Ctc/postgres

```
(8 rows)

\q
root@regress-stand:~#
```

4.3.4 Установка и настройка плагина для интеграции VMS с FindFace

1. Откройте файл `/opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py` и найдите раздел DATABASES. Запомните значение PASSWORD для пользователя ntech:

```
sudo vi /opt/findface-multi/configs/findface-multi-legacy/findface-multi-
legacy.py

...
# camera groups, watchlists and so on. Only PostgreSQL is supported.
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'DISABLE_SERVER_SIDE_CURSORS': True,
        'NAME': 'ffsecurity', 'HOST': '127.0.0.1', 'PORT': 5439, 'USER':
        'ntech', 'PASSWORD': 'PASSWORD'
    }
}
```

2. Скачайте установочный файл плагина FindFace (.deb) по ссылке от представителя NtechLab.
3. Установите его с помощью команды:

```
sudo dpkg -i findface-vms-integration-plugin_0.1.0_amd64.deb
```

4. Откройте конфигурационный файл расположенный в `/etc/findface-vms-integration-plugin.conf`. Раскомментируйте и вставьте в него следующие данные:

```
sudo vi /etc/findface-vms-integration-plugin.conf
```

- POSTGRES_SERVER (укажите сервер, на котором запущен контейнер postgresql)

- POSTGRES_PORT (используйте значение, отличное от указанного в config.py, например, используйте 5432)
- POSTGRES_USER (укажите ntech)
- POSTGRES_PASSWORD (используйте значение PASSWORD для пользователя ntech из конфигурационного файла /opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py, см. *n.1.*)
- POSTGRES_DB (укажите ffsintegration)
- PUBLIC_URL (по умолчанию <http://127.0.0.1:18333>)
- FFS_PUBLIC_URL (по умолчанию: <http://127.0.0.1>)
- FFS_TOKEN (укажите тот же токен, что в разделе EXTERNAL_VMS → TOKEN файла /opt/findface-multi/configs/findface-multi-legacy/findface-multi-legacy.py без префикса VmsPluginToken)

В результате конфигурационный файл будет выглядеть следующим образом:

Совет: В нашем примере мы устанавливаем плагин на тот же хост, где развернут FindFace Multi. В случае использования отдельного компьютера, пожалуйста, укажите правильный адрес сервера FindFace Multi в поле POSTGRES_SERVER, а URL – в поле FFS_PUBLIC_URL.

Остальные параметры являются необязательными и не требуются для первоначальной настройки.

5. Обновите базу данных, используя следующую команду:

```
sudo /opt/findface-vms-integration-plugin/bin/vms-integration-plugin-alembic upgrade head
```

В случае успеха результат будет выглядеть следующим образом:

```
root@regress-stand:/home/presale# nano /etc/findface-vms-integration-plugin/alembic.ini
root@regress-stand:/home/presale# sudo /opt/findface-vms-integration-plugin/bin/vms-integration-plugin-alembic upgrade head
INFO [alembic.runtime.migration] Context impl PostgresqlImpl.
INFO [alembic.runtime.migration] Will assume transactional DDL.
INFO [alembic.runtime.migration] Running upgrade -> 809ccbb29324, init
root@regress-stand:/home/presale#
```

6. Чтобы запустить сервис плагина и убедиться, что сервис запускается автоматически, выполните следующую команду:

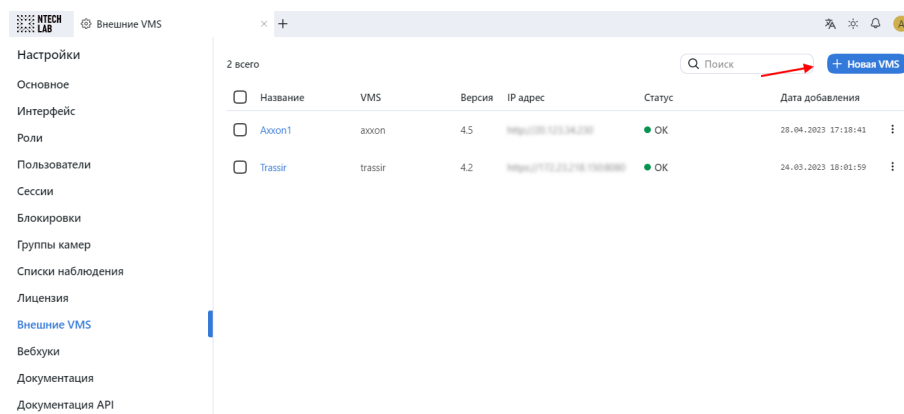
```
sudo systemctl start findface-vms-integration-plugin && sudo systemctl enable findface-vms-integration-plugin && sudo systemctl status findface-vms-integration-plugin
```

```
root@regress-stand:~# systemctl start findface-vms-integration-plugin
root@regress-stand:~# systemctl enable findface-vms-integration-plugin
Created symlink /etc/systemd/system/multi-user.target.wants/findface-vms-integration-plugin.service → /lib/systemd/system/findface-vms-integration-plugin.service.
root@regress-stand:~#
```

Предупреждение: Для корректного воспроизведения видеоархивов с внешних VMS необходимо установить на внешнем сервере VMS тот же часовой пояс, что и на сервере FindFace Multi.

4.3.5 Добавление Внешних VMS в FindFace Multi

1. Войдите в FindFace Multi в вашем браузере, перейдите в раздел *Настройки* -> *Внешние VMS*. Нажмите на кнопку *Новая VMS*.



2. Укажите параметры вашей VMS в открывшихся окнах и нажмите на кнопку *Сохранить*, как только все будет готово:

Trassir

Сведения Камеры Полученные сообщения Отправка сообщ...

Название
Trassir

VMS: trassir Версия: 4.2

Trassir API URL (HTTPS)
https://192.168.1.100

Trassir API Username
findface

Trassir API Password

Trassir SDK Password

Camera URL type (api, rtsp)
api

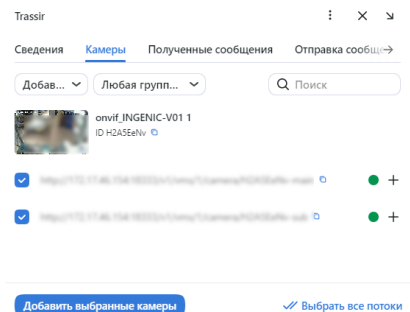
Trassir event name
face_match

Trassir event thumbnail (event_fullframe, card_thumbnail)
event_fullframe

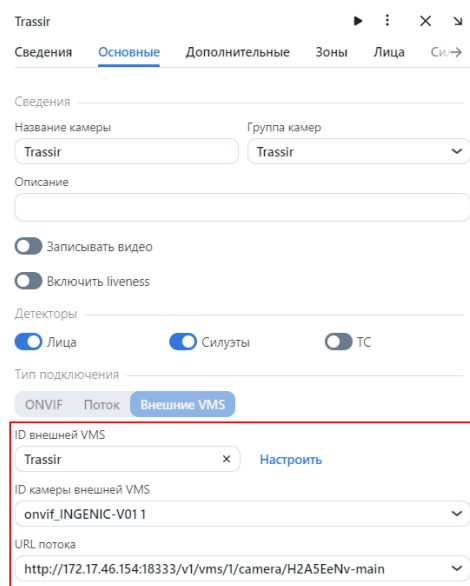
Trassir video timeline event time delta (0-10s, default 2s)
5

Event types that must be excluded from event subscription (comma separated)

3. После завершения теперь вы можете управлять потоками со своих VMS на вкладке *Камеры*. Например, вы можете мгновенно добавить все или выбранные камеры в FindFace Multi.



4. Если вам нужно отправить события на внешнюю VMS, в настройках камеры, добавленной из VMS, проверьте, настроена ли отправка событий на внешние VMS:



5. В обязательных свойствах списка наблюдения установите флажок **Отправка событий сопоставления в VMS** и нажмите на кнопку *Сохранить*, чтобы применить изменения.

6. Отправленные и полученные сообщения можно просмотреть на соответствующих вкладках выбранной VMS:

Теперь ваша установка FindFace Multi расширена и поддерживает внешние VMS. Вы можете подключить столько VMS, сколько потребуется, и легко добавлять существующие видеопотоки в FindFace всего за несколько кликов.

4.4 Внешние детекторы

Существует возможность интеграции FindFace Multi с внешними детекторами, предоставляющими кадры для распознавания объектов, например, с терминалами контроля доступа. В этом случае при получении кадра от внешнего детектора FindFace Multi будет автоматически инициировать извлечение вектора признаков объекта и создание события. Работа с данными событиями выполняется по аналогии с *событиями* с камер видеонаблюдения.

Важно: Чтобы использовать внешний детектор для распознавания объектов, необходимо включить определение витальности лиц. Если вы не включили определение витальности лиц при *установке* продукта, можно сделать это сейчас, воспользовавшись информацией из раздела *Включение определения витальности лиц*.

Модель для определения витальности `liveness.web.v0`, включенная в конфигурацию по умолчанию, используется для аутентификации по лицу. Если ваш сценарий использования подразумевает подключение к внешнему детектору для распознавания объектов, замените модель `liveness.web.v0` на

модель `liveness.pacs.v2` в файле конфигурации `findface-extraction-api.yaml`. Так как обе модели используют один и тот же нормализатор `facenorm.multicrop_full_crop2x_size400.cpu.fnk`, менять его не нужно.

Замените модель для `liveness` в файле конфигурации `/opt/findface-multi/configs/findface-extraction-api/findface-extraction-api.yaml`:

GPU

```
sudo vi /opt/findface-multi/configs/findface-extraction-api/findface-extraction-api.yaml

extractors:
  ...
  models:
    face_liveness: faceattr/liveness.pacs.v2.gpu.fnk
  ...
```

CPU

```
sudo vi /opt/findface-multi/configs/findface-extraction-api/findface-extraction-api.yaml

extractors:
  ...
  models:
    face_liveness: faceattr/liveness.pacs.v2.cpu.fnk
  ...
```

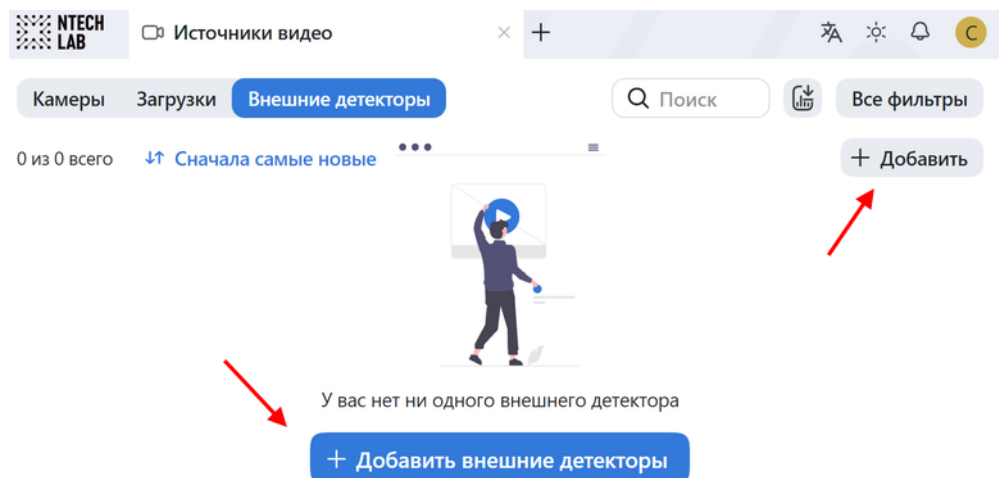
Перезапустите контейнер `findface-multi-findface-extraction-api-1`.

```
sudo docker container restart findface-multi-findface-extraction-api-1
```

Интеграция внешних детекторов выполняется через *HTTP API*. После первичной настройки FindFace Multi выдаст токен. Указывайте данный токен в каждом API-запросе, отправляемом внешним детектором в FindFace Multi, чтобы авторизовать устройство.

Для интеграции внешнего детектора с FindFace Multi выполните следующие действия:

1. Перейдите в *Источники Видео -> Внешние детекторы*
2. Нажмите *+ Добавить* или *+ Добавить внешние детекторы*, если еще не добавлено ни одного детектора.



3. В открывшейся вкладке добавьте внешний детектор в группу камер, чтобы впоследствии было удобнее фильтровать события с данного устройства.

Устройства Обнаружение ONVIF Загрузка файла **Внешние детекторы** ✕

Внешний детектор

Название внешнего детектора Исходная группа камер

Детектор 1 Default Camera Group ▼

Описание

Вход

Добавить и настроить > Добавить >

Совет: Вы можете выделить отдельную группу камер специально под внешние детекторы.

4. На вкладке *Добавить и настроить* включите liveness. На этой же вкладке вы увидите токен.

Детектор 1 Не активна



Название внешнего детектора

Детектор 1

Исходная группа камер

Default Camera Group



Описание

Вход

☒ Включить liveness

Токен



ID 9

Создано 28.03.2023 10:38:54

Обновлено 28.03.2023 10:38:54

Сохранить

Отменить

5. Указывайте данный токен в каждом API-запросе, отправляемом внешним детектором в FindFace Multi для создания события. В результате кадры, переданные в запросах, будут связаны с камерой, соответствующей внешнему детектору, и обработаны по аналогии с кадрами с камер видеонаблюдения.

FindFace Security API doc

Filter Search

GET /events/bodies/{id}/
 PATCH /events/bodies/{id}/
 POST /events/bodies/acknowledge/
 POST /events/bodies/add/
 GET /events/cars/
 GET /events/cars/{id}/
 PATCH /events/cars/{id}/
 POST /events/cars/acknowledge/
 POST /events/cars/add/
 GET /events/faces/
 GET /events/faces/{id}/
 PATCH /events/faces/{id}/
 POST /events/faces/acknowledge/
 POST /events/faces/add/

Create face events from provided image

POST /events/faces/add/

Use this method to create new face events.

REQUEST

REQUEST BODY* multipart/form-data

token string Min 1 chars
 Events creation api token

fullframe binary
 Выберите файл | файл не выбран
 Full frame of event

rotate boolean
 Try to rotate source

camera integer
 Related camera

timestamp date-time | null
 Timestamp of event

Подробная интерактивная документация HTTP API FindFace Multi доступна после установки по адресу http://<ffsecurity_ip:port>/api-docs. Изучайте и пробуйте.

Совет: Документацию также можно найти в веб-интерфейсе, перейдя в меню по пунктам *Настройки* -> *Документация API*.
