
FindFace Multi

Выпуск 1.2

NtechLab

янв. 16, 2024

1	Новое в FindFace Multi 1.2	5
2	Руководство системного администратора	9
2.1	Архитектура	9
2.2	Требования	16
2.3	Лицензирование	21
2.4	Развертывание FindFace Multi	24
2.5	Основные настройки	53
2.6	Управление пользователями и безопасность системы	55
2.7	Настройка расширенного функционала	73
2.8	Обслуживание и устранение неисправностей	110
2.9	Приложения	148
3	Руководство пользователя	195
3.1	Приступая к работе	195
3.2	Основы веб-интерфейса	199
3.3	Управление видеокамерами. Видеомегнитофон	201
3.4	Настройка мониторинга объектов и картотеки	213
3.5	Счетчики лиц, силуэтов и машин. Определение дистанции	228
3.6	Управление областями	237
3.7	События и эпизоды распознавания объектов	243
3.8	Идентификация объектов в офлайн видео	254
3.9	Поиск объектов в базах данных	261
3.10	Сравнение двух объектов	263
3.11	Кластеры лиц, силуэтов, автомобилей	264
3.12	Связанная с людьми аналитика	274
3.13	Отчеты	278
3.14	Видеостена	284
4	Руководство по интеграции	287
4.1	HTTP API	287
4.2	Вебхуки	287
4.3	Интеграции с партнерами	297
4.4	Периферийные устройства	316
	Содержание модулей Python	319

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.
- Автомобили, 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 человека, возраст, автомобильный номер 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 функционал распознавания объектов.

Ключевые функции

- Платформа на основе AI.
- Комплексная картотека, содержащая сборную информацию о человеке/транспортном средстве. Возможность создания связей между карточками одного или разных типов.
- Быстрый и надежный мониторинг видео в реальном времени по картотеке.
- Мультиобъектная идентификация и аналитика: лица, силуэты, автомобили.
- 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 Видеомегнитофон).
- Мультиобъектная верификация: лица, силуэты, автомобили.
- 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 автомобиля.
- Поиск лиц, силуэтов, автомобилей в базах данных.

- Возможность подсчета лиц, силуэтов и автомобилей на подключенных камерах и определения расстояния между силуэтами. Поддержка подсчета по одной и нескольким камерам.
- Видеонаблюдение. Запись видео с камер (функционал Видеомегнитофон).
- 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 автомобиля), 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.
- Мобильное приложение.

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.
- Аналитика «Знай своего клиента» (KYC).
- Детализированные отчеты на события распознавания лиц, эпизоды, события поиска, кластеры, камеры, карточки, журнал действий пользователей, аналитику KYC и области.
- 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 1.2

Новые функции:

- **Видеомагнитофон** (бета-версия): новая функция, позволяющая записывать, хранить и проигрывать видеоданные с камер.

См.:

- *Архитектура*
- *Развертывание из консольного инсталлятора*
- *Пошаговое развертывание функционала Видеомагнитофон*
- *Настройка Видеомагнитофона*
- *Включение видеозаписи для камеры*
- *Видеоплеер в уведомлении о событии*

- **Картотека:** закономерный результат развития базы данных досье. В картотеке присутствуют карточки двух типов. Карточка человека содержит фотографии его лица и силуэта. Фотографии транспортных средств и сведения о регистрационных номерах хранятся в автомобильных карточках.

См. *Настройка мониторинга объектов и картотеки.*

- **Связи:** обогатите свою картотеку и возможности аналитики, настроив связи между карточками одного типа или разных типов.

См. *Связи карточки.*

- **Спецтехника:** новая функция на основе AI, которая позволяет автоматически распознавать транспортные средства, принадлежащие полиции, скорой помощи, службам спасения и такси.

См. *Распознавание автомобилей и их атрибутов.*

- **Счетчики:** возможность подсчета автомобилей в дополнение к лицам и силуэтам.

См. *Счетчики лиц, силуэтов и машин. Определение дистанции.*

- **Кластеры:** полностью переработанная функциональность Персон. Включает в себя автоматическую кластеризацию объектов одного происхождения (изображения лица/силуэта, принадлежащие одному человеку; изображения одного автомобиля), сквозную интеграцию с картотекой и многие другие возможности.

См. *Кластеры лиц, силуэтов, автомобилей*.

- **Эпизоды:** важные улучшения, связанные с разделением эпизодов на два типа: люди и автомобили.

См. *Эпизоды событий*.

- **Вебхуки:** возможность настройки вебхуков для областей и новых типов эпизодов.

См. *Вебхуки*.

- **Отчеты:** возможность построения отчетов по областям, активациям областей и записям активаций.

См. *Отчеты*.

- **Отчеты:** новые отчеты о рабочем времени с возможностью экспорта в Excel. Позволяют фиксировать точные моменты входа и выхода персонала на предприятие или в заданную область и подсчитывать общее время, проведенное в них. Могут быть построены на вкладках *События* и *Области*.

См. *Отчеты*.

Улучшенные алгоритмы, интерфейс, удобство использования:

- *Определение расстояния между силуэтами.*
- *Анализ взаимодействий.*
- *События.*
- *Области.*
- Расширено использование индикаторов выполнения процессов в веб-интерфейсе.
- *Вебхуки.*

Технические изменения:

- *Резервное копирование и восстановление FindFace Multi и всех данных.*
- *Скрипт удаления экземпляра продукта.*
- Для экономии места на диске система теперь сохраняет нормализованные изображения в формате JPEG с качеством 95% вместо PNG.
- *Видеообработка:* статистика теперь разбивается по типам объектов.
- *Утилита очистки системы:* поддержка удаления новых сущностей.
- *Пользовательские метаданные в Tarantool.*
- *Миграция кластеров* на другую нейронную сеть.

Переименования:

- `dossier` → `card`.
- *Связи* → *Анализ взаимодействий*.
- *Персоны* → *Кластеры*.

Новые нейронные сети:

- Распознавание специальных транспортных средств: новая нейронная сеть `carattr.special_types.v0` для распознавания полицейских автомобилей, скорой помощи, служб спасения и такси.
- Распознавание живых лиц (`liveness`): новые нейронные сети `liveness.pacs.v0` (встроена в `findface-video-worker`) и `liveness.colombo` (`findface-liveness-api`) с улучшенными характеристиками.
- Нормализатор изображений автомобильных номеров: `anaferon.v3` с улучшенными характеристиками.
- Распознавание автомобильных номеров: новая версия `license_plate.v4`, поддерживающая номерные знаки Узбекистана, Бразилии и Индии в дополнение к ОАЭ, России, Казахстану, Грузии, ЮАР, Вьетнаму, Белоруссии, Украине, Армении, Кыргызстану.

В следующем релизе:

- **Распределенная картотека:** возможность распределения картотеки по нескольким серверам с синхронизацией и репликацией.
- **Гибкая структура базы векторов признаков:** возможность создавать разные метаполя в разных галереях Tarantool.

Примечание: Данная функциональность уже доступна в FindFace Core. Если функциональность нужна вам сейчас, обратитесь в нашу службу поддержки (support@ntechlab.com).

Руководство системного администратора

Эта глава посвящена разворачиванию и последующему обновлению и техническому обслуживанию FindFace Multi 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*
- *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
- автомобили

Примечание: Функционал распознавания лиц включен по умолчанию. Внесите изменения в файлы конфигурации, чтобы включить распознавание *силуэтов* и *автомобилей*.

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 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 Видеомегнитофон — дополнительный функционал, который записывает, хранит и проигрывает видеоданные с камер.



Ядро FindFace

Ядро FindFace включает в себя следующие компоненты:

Ком- по- нент	Ис- поль- зуемые порты	Описание	По- став- щик
findface-extraction-api	8666	Сервис, использующий нейронные сети для обнаружения объекта на изображении и извлечения его вектора признаков. Он также распознает атрибуты объектов (например, пол, возраст, эмоции, борода, очки, медицинская маска для объектов типа лицо). CPU- или GPU-ускорение.	Собственная разработка NtechLab
findface-sf-api	8411	Сервис, реализующий HTTP API обнаружения и распознавания объектов.	
findface-tarantool-server	32001, 32002, 330xx, 81xx)	Сервис, обеспечивающий взаимодействие между сервисом findface-sf-api и базой данных векторов признаков (базой, в которой хранятся векторы признаков объектов) на основе Tarantool.	
findface-upload	8333	Веб-сервер на базе NginX, используемый как хранилище исходных изображений, миниатюр объектов и нормализованных изображений объектов. Если установлен Видеомегнитофон, findface-upload также хранит видеоданные с камер.	
findface-facerouter	8820	Сервис, который используется для задания правил обработки обнаруженных на видео объектов. В FindFace Multi функции findface-facerouter выполняются findface-security (см. <i>Прикладной модуль FindFace Multi</i>). Однако в целях интеграции вы можете установить и настроить данный компонент (см. <i>findface-facerouter</i> и <i>пользовательские плагины</i>).	
findface-video-manager	8810, 18811	Сервис, являющийся частью модуля видеодетекции объектов, через который осуществляется управление детекцией объектов на видео, а именно задаются настройки и список видеопотоков для обработки.	
findface-video-worker	8999	Сервис, часть модуля видеодетекции объектов, который распознает объекты на видео и отправляет их нормализованные изображения, соответствующие видеокadres и метаданные (такие как время обнаружения) в сервис findface-facerouter для дальнейшей обработки в соответствии с заданными правилами. Если включен <i>Видеомегнитофон</i> , findface-video-worker отправляет видео в сервис findface-video-storage . Обеспечивает <i>распознавание живых лиц</i> при соответствующих настройках. Поддерживается CPU- и GPU-ускорение.	
findface-ntls	443 (TCP), 3133, 3185	Локальный сервер лицензий, который проверяет подлинность <i>лицензии</i> FindFace Multi, взаимодействуя с глобальным сервером лицензий NtechLab. Для закрытых систем поддерживается лицензирование через ключ аппаратной защиты и по слепку с оборудования. Поддерживается лицензирование через прокси-сервер.	
findface-counter	8300	Сервис, используемый для дедупликации событий.	
findface-liveness-api	8301	Помимо встроенного функционала, предоставляемого findface-video-worker , распознавание живых лиц (liveness) также предоставляется автономным сервисом findface-liveness-api . Сервис берет определенное количество кадров из видеотрегмента и возвращает изображение лица наилучшего качества и результат liveness в десятичном виде, усредненный по взятым кадрам (см. <i>Liveness как автономный сервис</i>). Сервис findface-liveness-api также участвует в процессе <i>аутентификации пользователя по лицу</i> .	
2.1. Архитектура	Тарантул шар-дов (по	Стороннее программное обеспечение, на основе которого реализована база данных, хранящая извлеченные векторы признаков и события распознавания объектов. Системные данные, карточки, пользовательские аккаунты и настройки камер хранятся в PostgreSQL (часть прикладного модуля FindFace	13 Tarantool

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- security	На- страи- ва- ет- ся	Компонент, обеспечивающий доступ конечного пользователя к функциям ядра FindFace. Обеспечивает взаимодействие между ядром FindFace и веб-интерфейсом, функционирование системы в целом, HTTP и веб-сокет, обновление содержимого базы векторов признаков, отправку уведомлений о событиях, объединение событий в эпизоды, работу вебхуков, счетчики, кластеризацию объектов, постановку в очередь видеофайлов на обработку. Включает в себя следующие внутренние сервисы: проверка лицензии, менеджер счетчиков, менеджер вебхуков, кластеризатор, менеджер эпизодов, менеджер очереди на обработку видеофайлов. Последние четыре сервиса можно включать и выключать через файл конфигурации <code>/etc/findface-security/config.py</code> .	Соб- ствен- ная раз- ра- бот- ка NtechLab NATS
findface- security- ui	На- страи- ва- ет- ся	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 <code>findface-security</code> .	
etcd	2379	Стороннее программное обеспечение, реализующее блокировки в сервисе <code>findface-security</code> , 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 автомобилей, 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-пользу-емые пор-ты	Описание	По-став-щик
findface-video-storage	48611	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-security 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.	Собственная разработка NtechLab
findface-video-streamer-cpu	9000	По запросу от findface-security-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:

- *Подробно о компонентах*
- *Хранилища данных FindFace Multi*

2.1.3 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 (с одним установленным компонентом **findface-video-worker**).

См. раздел *Дополнительное развертывание findface-video-worker на удаленных серверах*.

- Полностью распределенная архитектура FindFace Multi. Может понадобиться балансировка нагрузки.

См. раздел *Стандартное развертывание в многосерверной среде*.

2.1.4 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*.

Если установка FindFace Multi выполняется из *apt-репозитория*, на CPU-сервере нужно развернуть пакеты `findface-extraction-api` и/или `findface-video-worker-cpu`, а на GPU-сервере пакеты `findface-extraction-api-gpu` и/или `findface-video-worker-gpu`.

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

Важно: Если разрешение используемой камеры превышает 1280x720 пикселей, настоятельно рекомендуется использовать пакет с ускорением на GPU `findface-video-worker-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
- Требования к камерам видеонаблюдения
 - Распознавание лиц
 - Распознавание силуэтов и автомобилей

2.2.1 Системные требования для базовой конфигурации

Для расчета характеристик серверов FindFace Multi используйте приведенные ниже требования.

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

Важно: Если разрешение используемой камеры 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 Multi требуется 2 ядра HT > 2.5 ГГц. Характеристики также зависят от количества камер. Для одной камеры 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 серии: GeForce (Maxwell, Pascal, Turing и выше), Tesla (Maxwell, Pascal, Volta v100, Turing 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 Multi требуется 8 Гб. Потребление памяти также зависит от количества камер. Для одной камеры 720p@25FPS требуется 2 Гб RAM	
HDD (SSD для лучшей производительности)	16 Гб	16+ Гб
	На собственные нужды операционной системы и FindFace Multi 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 18.04, только x64	

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

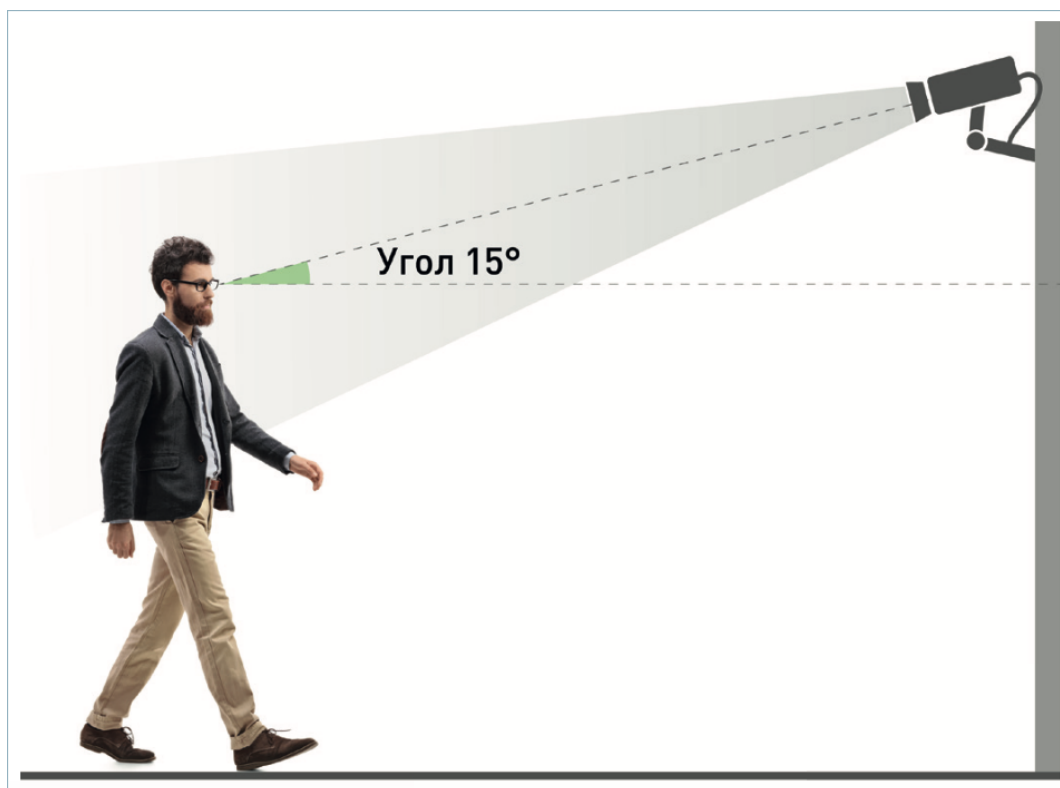
Администратор FindFace Multi должен обладать навыками работы с ОС Ubuntu на уровне продвинутого пользователя.

2.2.3 Требования к камерам видеонаблюдения

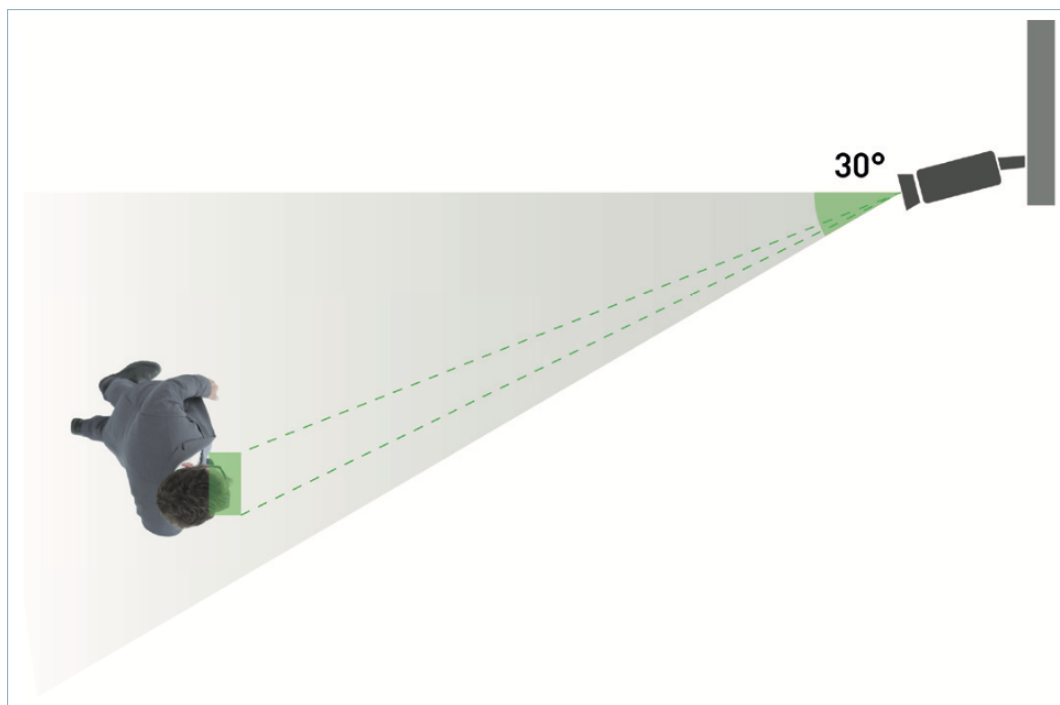
Распознавание лиц

К установке и характеристикам камер видеонаблюдения в системе распознавания лиц на основе FindFace Multi предъявляются следующие основные требования:

1. Для корректной детекции в видеопотоке установите камеру таким образом, чтобы в ее поле зрения обязательно появлялось лицо каждого человека, входящего в зону наблюдения.
2. Угол вертикального наклона видеокамеры не должен превышать 15° . Вертикальный наклон — это отклонение оптической оси видеокамеры от горизонтальной плоскости, расположенной на уровне середины лица человека среднего роста (160 см).



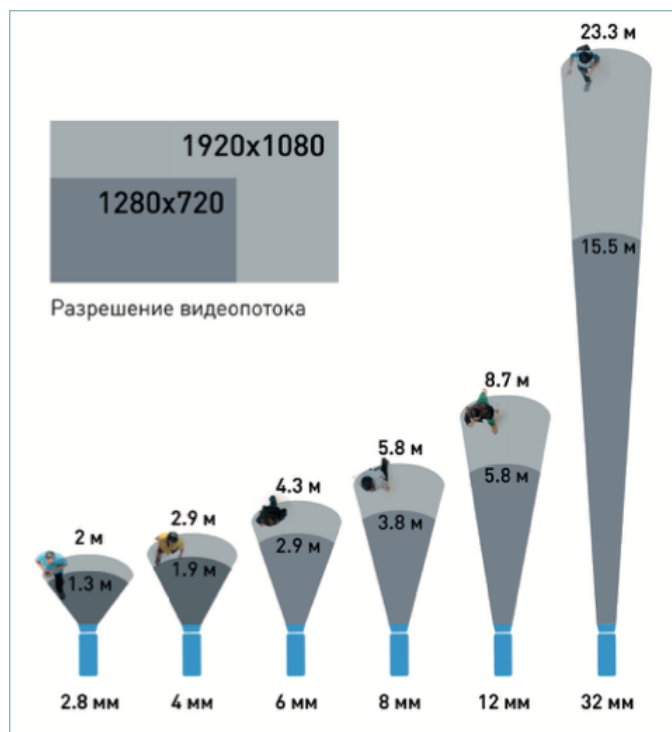
3. Угол горизонтального отклонения не должен превышать 30° . Горизонтальное отклонение — это отклонение оптической оси видеокамеры от вектора движения основного потока объектов распознавания.



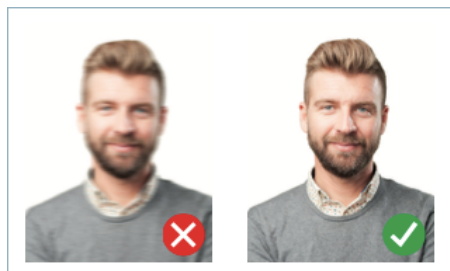
4. Минимальная плотность пикселей для идентификации – 500 пикселей/м (примерно соответствует ширине лица 80 пикселей).



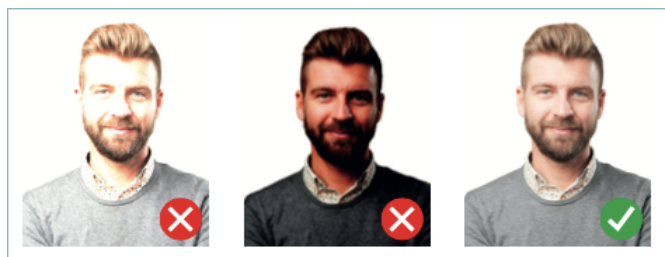
5. Фокусное расстояние объектива должно выбираться таким образом, чтобы при заданном расстоянии до объектов съемки обеспечивалась необходимая плотность пикселей. На рисунке ниже приведен пример расчета фокусного расстояния объектива от расстояния между камерой и объектами съемки. Для расчета фокусного расстояния для конкретной камеры требуется использовать калькуляторы или методологию, предоставляемые производителем камеры.



6. Экспозиция должна быть настроена таким образом, чтобы лица были резкими (“в фокусе”), не смазанными и равномерно освещенными (не засвеченными и не слишком темными).



7. В зависимости от условий освещения (яркая засветка, слишком яркое или слишком тусклое освещение) рекомендуется использовать камеры с аппаратным WDR (Wide Dynamic Range) или другими технологиями, обеспечивающими компенсацию встречной засветки и/или слабой освещенности (BLC, HLC, DNR, высокая светочувствительность, Smart ИК-подсветка, AGC и др.).



8. Сжатие видео: большинство видеоформатов и кодеков, которые могут быть декодированы

FFmpeg.

9. Протоколы передачи видеопотока: RTSP, HTTP.

Совет: Для расчета точной конфигурации оборудования в соответствии с вашими целями обратитесь к нашим специалистам по адресу support@ntechlab.com.

Распознавание силуэтов и автомобилей

Поскольку распознавание силуэтов и автомобилей — функции относительно новые, нам еще требуется провести дополнительное тестирование в полевых условиях, чтобы сформулировать требования, которые были бы универсальными для всех проектов. Однако мы всегда рады индивидуально помочь вам с выбором и развертыванием оборудования с учетом потребностей вашего бизнеса. Не стесняйтесь обращаться к нашим специалистам (support@ntechlab.com).

2.3 Лицензирование

В этой главе:

- *Принцип лицензирования*
- *Просмотр и обновление лицензии*
- *Лицензирование в закрытой сети по слепку с оборудования*

2.3.1 Принцип лицензирования

FindFace Multi лицензируется по следующим критериям:

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

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; 1 автомобиль = 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.

5. Распознавание атрибутов силуэтов: цвет/тип одежды.
6. Распознавание атрибутов автомобилей: марка/модель/цвет/тип кузова.
7. Распознавание автомобильного номера.
8. Распознавание живых лиц в реальном времени (Liveness).
9. Интеграции с партнерами.

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.

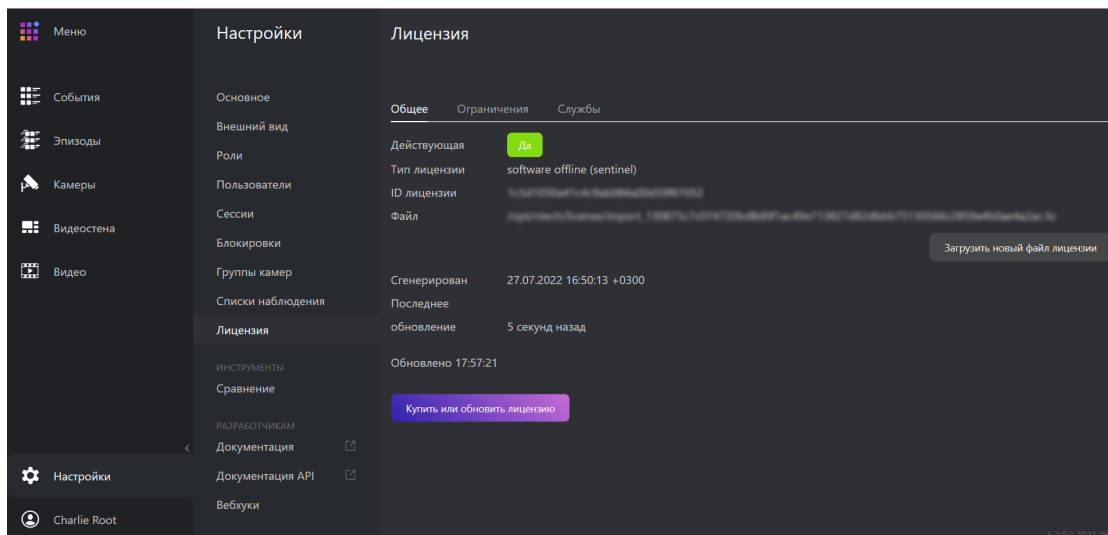
Примечание: Уведомите своего менеджера, если требуется продлить данный период (до 2-х дней).

- 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.

2.3.2 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.

2.3.3 Лицензирование в закрытой сети по слепку с оборудования

Примечание: Sentinel - вид оффлайн 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 оффлайн-новое 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 в системе лицензирование по слепку, выполните следующие действия:

1. Сообщите своему менеджеру о том, что намерены использовать данный метод лицензирования, и запросите свой уникальный идентификатор лицензии.
2. Установите драйверы Sentinel на физическом сервере с компонентом `findface-ntls`.

Выполните следующие действия:

1. Загрузите [драйверы Sentinel](#) с официального сайта.
2. Разархивируйте скачанный файл 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 серверов Sentinel.

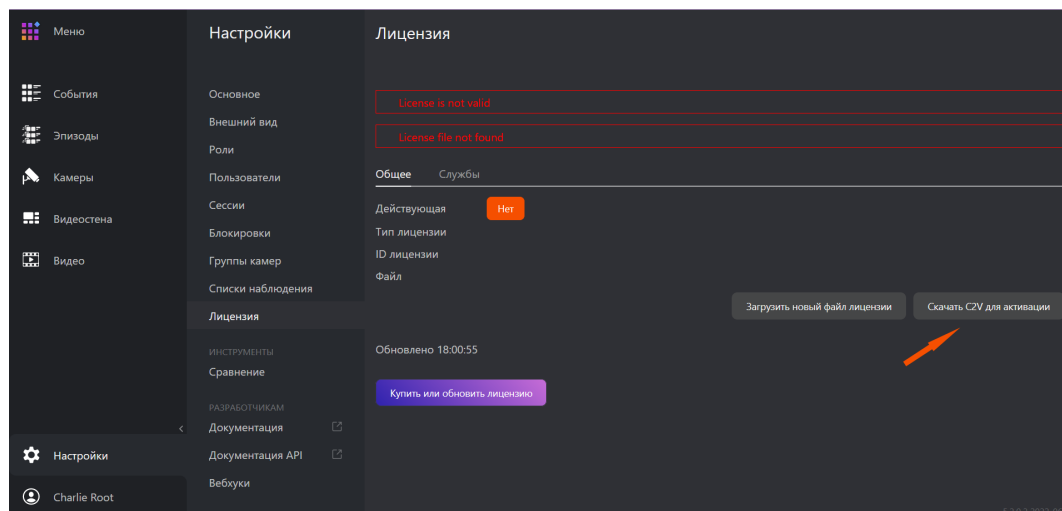
```
sudo systemctl start aksusbd.service hasplmd.service
sudo systemctl status aksusbd.service hasplmd.service
```

3. 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
```

4. Сгенерируйте файл C2V в FindFace Multi.

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 'http://<findface-ntls-server-ip>/ntls/c2v' >my_pc.c2v
```

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

2.4 Развертывание FindFace Multi

FindFace Multi предоставляет следующие варианты развертывания:

- развертывание из консольного инсталлятора
- пошаговое развертывание из apt-репозитория

Важно: Первый после развертывания запуск сервисов с GPU-ускорением `findface-extraction-api` и `findface-video-worker-gpu` 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.

2.4.1 Развертывание из консольного инсталлятора

Для развертывания FindFace Multi используется консольный инсталлятор.

Совет: Перед тем как приступить к развертыванию, обязательно ознакомьтесь с *системными требованиями*.

Важно: Если вы намерены использовать дополнительный функционал Видеомегнитофон и находиться на территории России или Беларуси, то перед тем, как приступить к развертыванию, необходимо обратиться в support@ntechlab.com для получения дистрибутива MongoDB. Далее следуйте инструкциям по установке в разделе *Устранение неполадок с дистрибутивом MongoDB*.

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*.

Для развертывания FindFace Multi из инсталлятора выполните следующие действия:

1. Загрузите файл инсталлятора `findface-multi-1.2-and-server-5.2.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-multi-1.2-and-server-5.2.run
```

4. Запустите файл `.run`.

```
sudo ./findface-multi-1.2-and-server-5.2.run
```

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

1. Устанавливаемый продукт: FindFace Multi.
2. Тип установки:
 - 1: установить FindFace Multi на одиночном физическом сервере.

- 2: установить FindFace Multi в качестве центрального сервера и настроить его на взаимодействие с дополнительными удаленными серверами `findface-video-worker`.

Совет: Для отдельной установки `findface-video-worker` см. *Дополнительное развертывание findface-video-worker на удаленных серверах*.

- 3: установить apt-репозиторий для *пошагового развертывания*.
- 4: *полностью настраиваемая установка*.

Примечание: Если вы выбрали тип установки №3 или №4, не забудьте установить необходимые модели нейронных сетей вместе с компонентом `findface-extraction-api`.

3. Тип пакета `findface-video-worker`: CPU или GPU.
4. Тип пакета `findface-extraction-api`: CPU или GPU.
5. Do you want to install NtechLab VMS?: yes (да) или no (нет), чтобы установить Видеомониторинг.

При выборе установки одиночного сервера FindFace Multi, его компоненты будут автоматически установлены, настроены и запущены в соответствии со следующей конфигурацией:

Важно: В случае чистой установки инсталлятор автоматически настроит `findface-extraction-api` на использование нейронной сети `kiwi_320`. В противном случае вам будет предложено сделать выбор между `kiwi_320` и предыдущей моделью. Категорически не рекомендуется использовать инсталлятор для обновления системы. См. инструкции по обновлению в разделе *Обновление до FindFace Multi 1.2*.

Сер-вис	Конфигурация
postgresql-10	Устанавливается и запускается.
nats-server	Устанавливается и запускается.
etcd	Устанавливается и запускается.
pgbouncer	Устанавливается и запускается.
memcached	Устанавливается и запускается.
nginx	Устанавливается и запускается.
django	Устанавливается и запускается как веб-фреймворк для веб-интерфейса FindFace Multi.
findface-ntls	Устанавливается и запускается.
findface-tarantool-server	Устанавливается и 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-extraction-api	Устанавливается и запускается (CPU/GPU-ускорение).
findface-sf-api	Устанавливается и запускается.
findface-upload	Устанавливается.
findface-video-manager	Устанавливается и запускается.
findface-video-worker-*	Устанавливается и запускается (CPU/GPU-ускорение).
findface-data-*	Модели нейронных сетей для распознавания объектов и их атрибутов. Устанавливаются.
findface-security	Устанавливается и запускается.
findface-counter	Устанавливается и запускается.
findface-liveness-api	Устанавливается и запускается.
jq	Устанавливается. Используется для структурирования API-ответов от FindFace Multi в формате JSON.
python3-ntech.*	Внутренние и вспомогательные сервисы. Устанавливаются и запускаются.
findface-video-storage	Устанавливается и запускается (только вместе с Видеомagniтофоном).
findface-video-streamer-cpu	Устанавливается и запускается (только вместе с Видеомagniтофоном).
mongodb	Устанавливается и запускается (только вместе с Видеомagniтофоном).

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

Совет: Обязательно сохраните эти данные: они вам понадобятся.

```
#####  
#                               Installation is complete                               #  
#####  
- upload your license to http://172.20.77.17/#/license/  
- user interface: http://172.20.77.17/  
  superuser:      admin  
  password:       admin  
  documentation:  http://172.20.77.17/doc/
```

5. Укажите свою временную зону в файле конфигурации `/etc/findface-security/config.py` в формате `Регион/Страна/Город` или `Etc/GMT+Н`. Временная зона определяет время в отчетах, логах, именах артефактов FindFace Multi, таких как полные кадры и миниатюры событий, скриншоты счетчиков и т. п.

Совет: Лучший способ сделать это — скопировать и вставить часовой пояс из [таблицы](#) в Википедии.

```
sudo vi /etc/findface-security/config.py  
  
# time zone  
TIME_ZONE = 'America/Argentina/Buenos_Aires'
```

6. (Опционально) В том же файле конфигурации вы можете изменить язык веб-интерфейса по умолчанию.

```
# language code  
LANGUAGE_CODE = 'es-ar'
```

См.также:

Добавление пользовательского языка

7. Перезапустите сервис `findface-security`.

```
sudo systemctl restart findface-security.service
```

8. 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 админи-

стратора от Супер Администратора в том, что последний не может лишиться прав администратора даже при смене роли.

9. Ответы на вопросы инсталлятора были сохранены в файл `/tmp/<findface-installer-*>.json`. Вы можете отредактировать его и использовать для установки FindFace Multi на других серверах, не отвечая на вопросы повторно.

Для этого выполните следующую команду:

```
sudo ./findface-multi-1.2-and-server-5.2.run -f /tmp/<findface-installer-*>.json
```

Совет: Пример данного файла можно посмотреть в разделе *Файл с параметрами установки*.

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 *Логи сервисов*.

Примечание: Для того чтобы настроить Видеомегнитофон после развертывания, следуйте *инструкции*.

2.4.2 Базовое пошаговое развертывание из арт-репозитория

Данный раздел содержит подробную информацию о базовом пошаговом развертывании компонентов FindFace Multi. Выполните приведенные ниже инструкции, придерживаясь заданного порядка.

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

Важно: В данном разделе не рассматривается развертывание функционала Видеомегнитофон. Вы можете найти соответствующую пошаговую инструкцию *здесь*.

Важно: Если находитесь на территории России или Беларуси и намерены использовать функционал Видеомagniтофон, то, перед тем, как приступить к развертыванию, необходимо обратиться в support@ntechlab.com для получения дистрибутива MongoDB. Далее следуйте инструкциям по установке в разделе *Устранение неполадок с дистрибутивом MongoDB*.

В этом разделе:

- Установка apt-репозитория
- Установка необходимого стороннего ПО
- Обеспечение лицензирования
- Развертывание основной базы данных
- Развертывание ядра FindFace
- Развертывание прикладного модуля FindFace Multi и базы данных векторов признаков

Установка apt-репозитория

Прежде всего установите apt-репозиторий FindFace следующим образом:

1. Загрузите файл инсталлятора `findface-multi-1.2-and-server-5.2.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-multi-1.2-and-server-5.2.run
```

4. Запустите файл `.run`.

```
sudo ./findface-multi-1.2-and-server-5.2.run
```

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

1. Устанавливаемый продукт: **FindFace Multi**.
2. Тип установки: `repo: Don't install anything, just set up the APT repository`.
3. Устанавливаемые модели нейронных сетей (при необходимости). Для того чтобы выбрать модели, сначала снимите выделение, введя в командной строке `-*`, затем введите через пробел порядковые номера нужных моделей, например: `1 3 4`. Введите `done` для сохранения выбора и перехода к следующему шагу.

Важно: Необходимо установить модель детектора хотя бы для одного типа объекта (лицо, тело или автомобиль).

После этого apt-репозиторий FindFace будет автоматически установлен.

Установка необходимого стороннего ПО

Для работы FindFace Multi необходимо стороннее программное обеспечение PostgreSQL, PgBouncer, NATS, etcd и memcached. Выполните следующие действия:

1. Установите пакеты с указанным сторонним ПО следующим образом:

```
sudo apt update
sudo apt install -y postgresql-10 nats-server etcd memcached pgbouncer
```

2. Откройте файл конфигурации `/etc/memcached.conf`. Установите максимальный размер памяти в мегабайтах, используемый для хранения элементов `memcached`: `-m 1024`. Установите максимальный размер элемента: `-I 16m`. Если один или оба этих параметра отсутствуют, добавьте их в файл.

```
sudo vi /etc/memcached.conf

-m 1024
-I 16m
```

3. Предоставьте надежный пароль пользователю `ntech` (`9T3g1nXy9yx3y8MIGm9fbef3dia8UTc3` в примере ниже). Внесите учетные данные в список пользователей `pgbouncer`.

```
echo "ntech" "9T3g1nXy9yx3y8MIGm9fbef3dia8UTc3" | sudo tee -a /etc/pgbouncer/
↪userlist.txt
```

4. Настройте `pgbouncer`. Откройте файл `/etc/pgbouncer/pgbouncer.ini` и добавьте базу данных `ffsecurity` в раздел `databases`. Настройте указанные параметры, как показано в примере ниже. Остальные параметры должны быть закомментированы.

```
sudo vi /etc/pgbouncer/pgbouncer.ini

[databases]
ffsecurity = dbname=ffsecurity host=localhost port=5432 user=ntech
[pgbouncer]
pidfile = /var/run/postgresql/pgbouncer.pid
listen_addr = 127.0.0.1
listen_port = 5439
unix_socket_dir = /var/run/postgresql
auth_type = plain
auth_file = /etc/pgbouncer/userlist.txt
pool_mode = transaction
server_reset_query = DISCARD ALL
max_client_conn = 16384
default_pool_size = 20
syslog = 1
```

5. Добавьте сервисы стороннего ПО в автозагрузку Ubuntu и перезапустите их:

```
sudo systemctl enable postgresql@10-main.service nats-server etcd.service memcached.
↪service pgbouncer.service
sudo systemctl restart postgresql@10-main.service nats-server etcd.service
↪memcached.service pgbouncer.service
```

Обеспечение лицензирования

Важно: См. *Лицензирование*, чтобы узнать о политике лицензирования NtechLab.

Для обеспечения лицензирования FindFace Multi разверните `findface-ntls`, сервер лицензий в составе ядра FindFace.

Важно: Система на базе FindFace Multi может включать в себя только один экземпляр `findface-ntls`.

```
sudo apt update
sudo apt install -y findface-ntls
sudo systemctl enable findface-ntls.service && sudo systemctl start findface-ntls.service
```

Развертывание основной базы данных

Основная база данных FindFace Multi построена на PostgreSQL. Для того чтобы развернуть основную базу данных, выполните следующие действия:

1. Откройте список пользователей `pgbouncer /etc/pgbouncer/userlist.txt`. Скопируйте пароль пользователя `ntech` (`9T3g1nXy9yx3y8MIGm9fbef3dia8UTc3` в примере ниже).

```
sudo cat /etc/pgbouncer/userlist.txt

"ntech" "9T3g1nXy9yx3y8MIGm9fbef3dia8UTc3"
```

2. В консоли PostgreSQL создайте пользователя `ntech` под скопированным паролем, а также базы данных `ffsecurity` и `ffcounter` PostgreSQL.

```
sudo -u postgres psql

postgres=# CREATE ROLE ntech WITH LOGIN PASSWORD '9T3g1nXy9yx3y8MIGm9fbef3dia8UTc3';

postgres=# CREATE DATABASE ffsecurity WITH OWNER ntech ENCODING 'UTF-8' LC_COLLATE=
↳ 'en_US.UTF-8' LC_CTYPE='en_US.UTF-8' TEMPLATE template0;

postgres=# CREATE DATABASE ffcounter WITH OWNER ntech ENCODING 'UTF-8' LC_COLLATE=
↳ 'C.UTF-8' LC_CTYPE='C.UTF-8' TEMPLATE template0;
```

Совет: Для выхода из консоли PostgreSQL введите `\q` и нажмите **Enter**.

3. Разрешите авторизацию в PostgreSQL по UID клиента сокета. Перезапустите PostgreSQL.

```
echo 'local all ntech peer' | sudo tee -a /etc/postgresql/10/main/pg_hba.conf

sudo systemctl restart postgresql@10-main.service
```


Развертывание ядра FindFace

Для развертывания ядра FindFace выполните следующие действия:

Совет: Вы можете найти описание компонентов ядра FindFace и их параметров конфигурации в разделах *Архитектура* и *Подробно о компонентах*.

1. Для FindFace Multi на GPU *установите драйверы NVIDIA*.

Важно: Обязательно перезагрузите сервер после завершения установки драйверов NVIDIA. В противном случае при последующей установке GPU-компонентов произойдет сбой.

2. Установите компоненты ядра FindFace:

На CPU:

```
sudo apt update
sudo apt install -y findface-tarantool-server findface-extraction-api findface-sf-
↪api findface-upload findface-video-manager findface-video-worker-cpu findface-
↪liveness-api
```

На GPU:

```
sudo apt update
sudo apt install -y findface-tarantool-server findface-extraction-api-gpu findface-
↪sf-api findface-upload findface-video-manager findface-video-worker-gpu findface-
↪liveness-api
```

Примечание: Если на физическом сервере установлено несколько видеокарт, см. *Использование нескольких видеокарт*.

Важно: Обязательно *установите* модели нейронных сетей на серверах с findface-extraction-api.

3. В файле конфигурации /etc/findface-sf-api.ini включите параметр allow-return-facen.

```
sudo vi /etc/findface-sf-api.ini

...
limits:
  ...
  allow-return-facen: true
...
```

4. Откройте файл конфигурации /etc/findface-video-worker-cpu.ini (/etc/findface-video-worker-gpu.ini). Задайте следующие параметры:

- В параметре `mgr-static` укажите IP-адрес сервера `findface-video-manager`, который будет обеспечивать `findface-video-worker` настройками и списком видеопотоков для обработки.
- В параметре `capacity` укажите максимальное количество видеопотоков, которое может быть обработано `findface-video-worker`.
- В разделе `streamer` укажите IP-адрес и порт для доступа к *видеостене*. Значение `port` должно быть 18999. Установите `tracks = true`, чтобы улучшить отображение рамок с объектами на видеостене.

```

sudo vi /etc/findface-video-worker-cpu.ini
sudo vi /etc/findface-video-worker-gpu.ini

mgr-static=127.0.0.1:18811

capacity=10

[streamer]
#-----
## streamer/shots webserver port, 0=disabled
## type:number env:CFG_STREAMER_PORT longopt:--streamer-port
port = 18999

## streamer url - how to access this worker on streamer_port
## type:string env:CFG_STREAMER_URL longopt:--streamer-url
url = 127.0.0.1:18999

## use tracks instead detects for streamer
## type:bool env:CFG_STREAMER_TRACKS longopt:--streamer-tracks
tracks = true

```

5. Добавьте сервисы ядра FindFace в автозагрузку Ubuntu и запустите их.

На CPU:

```

sudo systemctl enable findface-extraction-api findface-sf-api findface-video-
↪manager findface-video-worker-cpu findface-liveness-api
sudo systemctl start findface-extraction-api findface-sf-api findface-video-manager
↪findface-video-worker-cpu findface-liveness-api

```

На GPU:

```

sudo systemctl enable findface-extraction-api findface-sf-api findface-video-
↪manager findface-video-worker-gpu findface-liveness-api
sudo systemctl start findface-extraction-api findface-sf-api findface-video-manager
↪findface-video-worker-gpu findface-liveness-api

```

Развертывание прикладного модуля FindFace Multi и базы данных векторов признаков

Для развертывания прикладного модуля FindFace Multi, выполните следующие действия:

1. Установите компоненты `findface-security`, `findface-security-ui` и `findface-counter`. Добавьте сервис `findface-counter` в автозагрузку Ubuntu и запустите его.

```
sudo apt update
sudo apt install -y findface-security findface-security-ui findface-counter
sudo systemctl enable findface-counter && sudo systemctl start findface-counter
```

2. Перенесите схему базы данных из FindFace Multi в PostgreSQL, создайте с *предустановленные* пользовательские роли и первого администратора (т. н. Супер Администратора).

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

```
sudo findface-security migrate
sudo findface-security create_groups
sudo findface-security create_default_user
```

3. Создайте структуру базы данных векторов признаков на основе Tarantool, выполнив следующую команду.

```
sudo findface-security make_tnt_schema | sudo tee /etc/findface-security/tnt_schema.
↪lua
```

4. Откройте файл конфигурации `/etc/tarantool/instances.available/FindFace.lua`. Проверьте, содержит ли файл команду `dofile`, а также определения `meta_indexes` и `meta_scheme`, как показано в примере ниже. При необходимости измените файл.

```
sudo vi /etc/tarantool/instances.available/FindFace.lua

dofile("/etc/findface-security/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",
    meta_indexes=meta_indexes,
    meta_scheme=meta_scheme
})
```

Важно: IP-адрес и номер порта, указанные в секции `shards` файла конфигурации `/etc/findface-sf-api.ini`, должны быть идентичны указанным в секции `FindFace.start`.

```
sudo vi /etc/tarantool/instances.available/FindFace.lua

...

FindFace.start("127.0.0.1", 8101...)
```

```
sudo vi /etc/findface-sf-api.ini
```

(continues on next page)

(продолжение с предыдущей страницы)

```
storage-api:
...
shards:
- master: http://127.0.0.1:8101/v2/
...
```

Важно: Если вы внесли изменения в файл конфигурации `/etc/findface-sf-api.ini`, обязательно перезапустите сервис `findface-sf-api`:

```
sudo systemctl restart findface-sf-api.service
```

5. Добавьте сервис `findface-tarantool-server` в автозагрузку Ubuntu и запустите его.

```
sudo systemctl enable tarantool@FindFace.service && sudo systemctl start
↳tarantool@FindFace.service
```

6. Откройте файл конфигурации `/etc/findface-security/config.py`. Задайте следующие параметры:

Совет: Просмотреть исходную версию файла `/etc/findface-security/config.py` вы можете [здесь](#).

- **SERVICE_EXTERNAL_ADDRESS:** IP-адрес или URL FindFace Multi, являющимся приоритетным для Genetec и вебхуков. Если параметр не задан, система будет использовать для работы с данным функционалом значение `EXTERNAL_ADDRESS`.

Важно: Для использования Genetec и вебхуков обязательно укажите по крайней мере один из параметров: `SERVICE_EXTERNAL_ADDRESS`/ `EXTERNAL_ADDRESS`.

- **EXTERNAL_ADDRESS:** (опционально) IP-адрес или URL, который используется для доступа в веб-интерфейс FindFace Multi. Если этот параметр не указан вручную, система автоматически определяет его как внешний IP-адрес сервера.

Примечание: Для доступа в FindFace Multi вы можете использовать оба IP-адреса: как автоопределенный, так и указанный вручную в `EXTERNAL_ADDRESS`.

- **VIDEO_DETECTOR_TOKEN:** придумайте токен и укажите его в данном параметре, чтобы авторизовать модуль видеодетекции объектов.

Совет: Рекомендуется сгенерировать токен, выполнив следующую команду:

```
pwgen -sncy 50 1|tr " " "."
```

- **VIDEO_MANAGER_ADDRESS:** IP-адрес сервера `findface-video-manager`.
- **NTLS_HTTP_URL:** IP-адрес сервера `findface-ntls`.

- **ROUTER_URL**: IP-адрес сервера `findface-security`, который будет получать обнаруженные на видео объекты от экземпляров `findface-video-worker`. Адрес указывается внутренний или внешний, в зависимости от сети, в которой `findface-video-worker` взаимодействует с `findface-security`. Измените порт по умолчанию с учетом *настроек переадресации* с HTTP на HTTPS или вообще не указывайте его, оставив только IP-адрес.
- **SF_API_ADDRESS**: IP-адрес сервера `findface-sf-api`.
- **DATABASES** (раздел): заполните его следующим образом: `'PORT': 5439, 'USER': 'ntech', 'PASSWORD': '<пароль из /etc/pgbouncer/userlist.txt>'` (см. *Установка необходимого стороннего ПО*).

Совет: Если необходимо обеспечить безопасность данных, включите *SSL-шифрование*.

Совет: При необходимости установите `'IGNORE_UNMATCHED': True`, чтобы отключить запись события в базу данных, если обнаруженный объект отсутствует в списках наблюдения. Данную настройку рекомендуется использовать при большом количестве объектов.

7. Используя приведенную ниже команду, сгенерируйте ключ подписи для шифрования сессии (используется в Django) и задайте его в параметре `SECRET_KEY`.

```
pwgen -sncy 50 1|tr " " ". "
```

8. Запустите сервисы.

```
sudo systemctl enable findface-security
sudo systemctl start findface-security
```

9. Отключите сервер `nginx`, активный по умолчанию, и добавьте сервер `findface-security` в список включенных серверов. Перезапустите `nginx`.

```
sudo rm /etc/nginx/sites-enabled/default

sudo ln -s /etc/nginx/sites-available/ffsecurity-nginx.conf /etc/nginx/sites-
➔enabled/

sudo nginx -s reload
```

10. Обеспечение лицензирования:

- Обеспечьте лицензирование. Используйте веб-интерфейс FindFace Multi, чтобы *загрузить лицензионный файл*, который вы до этого получили от своего менеджера (*Настройки -> Лицензия*).
- При лицензировании в закрытой системе с помощью USB-ключа аппаратной защиты, вставьте его в USB-порт.
- Для лицензирования в закрытой системы по слепку с оборудования см. *Лицензирование в закрытой сети по слепку с оборудования*.

Важно: Для того чтобы войти в первый раз в систему, используйте учетную запись Суперадминистратора `admin:admin`, созданную по умолчанию.

Примечание: Для создания других пользователей или смены пароля Суперадминистратора см. раздел *Управление пользователями*.

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 *Логи сервисов*.

2.4.3 Дополнительное развертывание findface-video-worker на удаленных серверах

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 `/etc/findface-ntls.cfg` на сервере с `findface-ntls` и установите `listen = 0.0.0.0:3133`. Перезапустите `findface-ntls`.

```
sudo vi /etc/findface-ntls.cfg

## Address to accept incoming client connections (IP:PORT)
## type:string env:CFG_LISTEN longopt:--listen
listen = 0.0.0.0:3133
```

```
sudo systemctl restart findface-ntls.service
```

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

```
sudo vi /etc/findface-video-manager.conf

listen: 0.0.0.0:18810
...
```

(continues on next page)

(продолжение с предыдущей страницы)

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

```
sudo systemctl restart findface-video-manager.service
```

3. На сервере FindFace Multi откройте файл конфигурации `/etc/findface-security/config.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 /etc/findface-security/config.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*.

Совет: Если на сервере несколько видеокарт, перед разворачиванием `findface-video-worker-gpu` изучите раздел *Использование нескольких видеокарт*.

1. Загрузите файл инсталлятора `findface-multi-1.2-and-server-5.2.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-multi-1.2-and-server-5.2.run
```

4. Запустите файл `.run`.

```
sudo ./findface-multi-1.2-and-server-5.2.run
```

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

1. Устанавливаемый продукт: FindFace Video Worker.
2. Тип пакета `findface-video-worker`: CPU или GPU.
3. IP-адрес центрального сервера `findface-security`.

После этого процесс установки будет автоматически запущен.

Примечание: Если `findface-ntls` и/или `findface-video-manager` будут установлены на серверах, отличных от сервера `findface-security`, укажите их IP-адреса в файле конфигурации `/etc/`

findface-video-worker-cpu.ini (/etc/findface-video-worker-gpu.ini) после установки компонента.

```
sudo vi /etc/findface-video-worker-cpu.ini
sudo vi /etc/findface-video-worker-gpu.ini
```

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: Для того чтобы автоматически установить `findface-video-worker` на других серверах, не отвечая на вопросы инсталлятора, используйте файл `/tmp/<findface-installer-*>.json`. Запустите инсталлятор следующей командой:

```
sudo ./findface-multi-1.2-and-server-5.2.run -f /tmp/<findface-installer-*>.json
```

Пример файла `/tmp/<findface-installer-*>.json` можно посмотреть в разделе *Файл с параметрами установки*.

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 *Логи сервисов*.

2.4.4 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 подготовленный файл `findface-multi-1.2-and-server-5.2.run`.

```
sudo ./findface-multi-1.2-and-server-5.2.run
```

2. Тип установки: `Fully customized installation`.
3. 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.
4. Выберите модели для установки. После этого процесс установки будет автоматически запущен.

Вы можете найти установленные модели для распознавания объектов и их атрибутов в директории `/usr/share/findface-data/models/`. См. *Модели нейронных сетей*.

2.4.5 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.

Для запуска полностью настраиваемой установки нужно ответить на вопросы инсталлятора следующим образом:

- Устанавливаемый продукт: `FindFace Multi`.
- Тип установки: `Fully customized installation`.
- Устанавливаемые компоненты FindFace Multi: 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.
- Связанные вопросы, такие как тип ускорения: `CPU` или `GPU`.

2.4.6 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 Видеомagniтофон. 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`, 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*

Развертывание центрального сервера

Для развертывания центрального сервера FindFace Multi выполните следующие действия:

1. На выделенном физическом сервере *установите* FindFace Multi из инсталлятора следующим образом:
 - 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                               #
#####
- upload your license to http://172.20.77.17/#/license/
- user interface: http://172.20.77.17/
  superuser:      admin
  password:       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 `/etc/findface-ntls.cfg` и установите `listen = 0.0.0.0:3133`. Перезапустите `findface-ntls.service`.

```
sudo vi /etc/findface-ntls.cfg

## Address to accept incoming client connections (IP:PORT)
## type:string env:CFG_LISTEN longopt:--listen
listen = 0.0.0.0:3133
```

```
sudo systemctl restart findface-ntls.service
```

4. Разрешите доступ к сервису `findface-video-manager` с любого IP-адреса. Для этого откройте файл конфигурации `/etc/findface-video-manager.conf` и установите `listen: 0.0.0.0:18810` и `rpc:listen: 0.0.0.0:18811`. Перезапустите `findface-video-manager`.

```
sudo vi /etc/findface-video-manager.conf

listen: 0.0.0.0:18810
...
rpc:
  listen: 0.0.0.0:18811
```

```
sudo systemctl restart findface-video-manager.service
```

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-multi-1.2-and-server-5.2.run -f /tmp/<findface-installer-*>.
↪ json
```

Примечание: Если `findface-ntls` и/или `findface-video-manager` установлены на других серверах, чем `findface-security`, укажите их IP-адреса после установки в файле конфигурации `/etc/findface-video-worker-gpu.ini` (`/etc/findface-video-worker-cpu.ini`).

```
sudo vi /etc/findface-video-worker-cpu.ini
sudo vi /etc/findface-video-worker-gpu.ini
```

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
```

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` и `findface-data`. Для того чтобы их выбрать, сначала снимите выделение со всех компонентов, введя в командной строке `-*`, затем введите порядковые номера `findface-extraction-api` и `findface-data` через пробел. Введите `done` для сохранения выбора и перехода к следующему шагу.
- Тип ускорения `findface-extraction-api`: CPU или GPU.
- Необходимость в изменении файла конфигурации `/etc/findface-extraction-api.ini`: 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, автомобилей 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. *Модели нейронных сетей*, *Распознавание атрибутов лица*, *Распознавание автомобилей и их атрибутов*, *Распознавание силуэтов и их атрибутов*.

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-multi-1.2-and-server-5.2.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 установленным nginx.

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

Совет: Вы можете установить nginx следующим образом:

```
sudo apt update
sudo apt install nginx
```

2. На сервере шлюза создайте новый файл конфигурации Nginx.

```
sudo vi /etc/nginx/sites-available/extapi
```

3. Вставьте следующий текст в созданный файл конфигурации. 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 *распределенной сети* FindFace Multi.

```
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;
}
```

4. Включите балансировщик нагрузки в nginx.

```
sudo ln -s /etc/nginx/sites-available/extapi /etc/nginx/sites-enabled/
```

5. Перезапустите nginx.

```
sudo service nginx restart
```

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

```
sudo vi /etc/findface-extraction-api.ini

listen: 172.168.1.10:18666
```

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

```
sudo systemctl restart findface-extraction-api.service
```

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. Ответьте на вопросы инсталлятора следующим образом:
 - Устанавливаемый продукт: FindFace Multi.
 - Тип установки: Fully customized installation.
 - Устанавливаемые компоненты FindFace Multi: `findface-tarantool-server`. Для того чтобы его выбрать, сначала снимите выделение со всех компонентов, введя в командной строке `*`, затем введите порядковый номер `findface-tarantool-server` (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-multi-1.2-and-server-5.2.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 /etc/tarantool/instances.enabled/*shard* | grep -E ".start|(listen =)"`
```

Будет возвращен следующий результат:

```
listen = '127.0.0.1:33001',
FindFace.start("127.0.0.1", 8101, {
    listen = '127.0.0.1:33002',
FindFace.start("127.0.0.1", 8102, {
```

Номера портов указаны в секции FindFace.start: 8101, 8102 и т. д.

Настройка сетевого взаимодействия

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

1. Откройте файл конфигурации /etc/findface-sf-api.ini:

```
sudo vi /etc/findface-sf-api.ini
```

Задайте следующие параметры:

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

(continues on next page)

(продолжение с предыдущей страницы)

```
- master: http://172.168.1.13:8102/v2/
  slave: ''
```

2. Откройте файл конфигурации `/etc/findface-security/config.py`.

```
sudo vi /etc/findface-security/config.py
```

Задайте следующие параметры:

Параметр	Описание
<code>SERVICE_EXTERNAL_ADDRESS</code>	URL 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>	URL 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 /etc/findface-security/config.py

...
# SERVICE_EXTERNAL_ADDRESS prioritized for webhooks and genetec
SERVICE_EXTERNAL_ADDRESS = 'http://localhost'
EXTERNAL_ADDRESS = 'http://127.0.0.1'

...
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',
    ...
}
```

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 *Логи сервисов*.

2.4.7 Пошаговое развертывание функционала Видеомагнитофон

Внутренняя *архитектура* Ядра FindFace позволяет встроить Видеомагнитофон — дополнительный функционал, который записывает, хранит и проигрывает видеоданные с камер.

Существуют следующие способы развертывания функционала Видеомагнитофон:

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

Данный раздел посвящен пошаговому развертыванию функционала Видеомагнитофон.

Важно: Если находитесь на территории России или Беларуси и намерены использовать функционал Видеомагнитофон, то, перед тем, как приступить к развертыванию, необходимо обратиться в support@ntechlab.com для получения дистрибутива MongoDB. Далее следуйте инструкциям по установке в разделе *Устранение неполадок с дистрибутивом MongoDB*.

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

В этом разделе:

- Установка apt-репозитория
- Установка необходимого ПО MongoDB
- Установка основных компонентов

Установка apt-репозитория

Прежде всего установите apt-репозиторий FindFace следующим образом:

1. Загрузите файл инсталлятора `findface-multi-1.2-and-server-5.2.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-multi-1.2-and-server-5.2.run
```

4. Запустите файл `.run`.

```
sudo ./findface-multi-1.2-and-server-5.2.run
```

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

1. Устанавливаемый продукт: FindFace Multi.
2. Тип установки: `repo: Don't install anything, just set up the APT repository.`
3. Не устанавливайте модели нейронных сетей. Ответьте `no` на вопрос `APT repository doesn't include face recognition models. Do you want to install them now?`.

После этого apt-репозиторий FindFace будет автоматически установлен.

Установка необходимого ПО MongoDB

Для работы Видеомагнитофона требуется стороннее программное обеспечение MongoDB. Установите его следующим образом:

```
sudo apt update
sudo apt install -y mongodb
```

После установки сервис `mongodb` будет запущен и добавлен в автозагрузку Ubuntu. Вы можете проверить его статус, выполнив следующую команду:

```
sudo systemctl status mongodb

mongodb.service - An object/document-oriented database
Loaded: loaded (/lib/systemd/system/mongodb.service; enabled; vendor preset: enabled)
Active: active (running) since Mon 2022-05-16 10:23:04 UTC; 46s ago
...
```

Установка основных компонентов

Для того чтобы установить компоненты Видеомагнитофона, выполните следующие действия:

1. Установите сервисы `findface-video-storage` и `findface-video-streamer-cpu` из ранее установленного APT-репозитория.

```
sudo apt install -y findface-video-storage findface-video-streamer-cpu
```

После установки `findface-video-storage` будет запущен и добавлен в автозагрузку Ubuntu. Сервис `findface-video-streamer-cpu` будет просто установлен.

- Откройте файл конфигурации `/etc/findface-video-storage.conf`. Установите `webdav` в параметре `chunk-storage -> type`.

```
sudo vi /etc/findface-video-storage.conf

...
chunk-storage:
    type: webdav
```

- После сохранения изменений перезапустите `findface-video-storage`.

```
sudo systemctl restart findface-video-storage
```

- Запустите сервис `findface-video-streamer-cpu` и добавьте его в автозагрузку Ubuntu.

```
sudo systemctl enable findface-video-streamer-cpu && sudo systemctl start findface-
↪video-streamer-cpu
```

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

2.4.8 Добавление репозитория NVIDIA и установка драйверов (только GPU)

FindFace Multi на GPU требует предварительной установки драйверов NVIDIA.

Для добавления репозитория NVIDIA и установки драйверов выполните следующие действия:

Важно: Вам потребуется стабильное подключение к Интернету, так как загрузка пакетов драйверов будет происходить из веб-ресурса NVIDIA.

- Загрузите файл инсталлятора `findface-multi-1.1-and-server-5.1.run`.
- Поместите файл `.run` в любой каталог на сервере установки (например, `/home/username`).
- Из данного каталога сделайте файл `.run` исполняемым.

```
chmod +x findface-multi-1.1-and-server-5.1.run
```

- Запустите файл `.run`.

```
sudo ./findface-multi-1.1-and-server-5.1.run
```

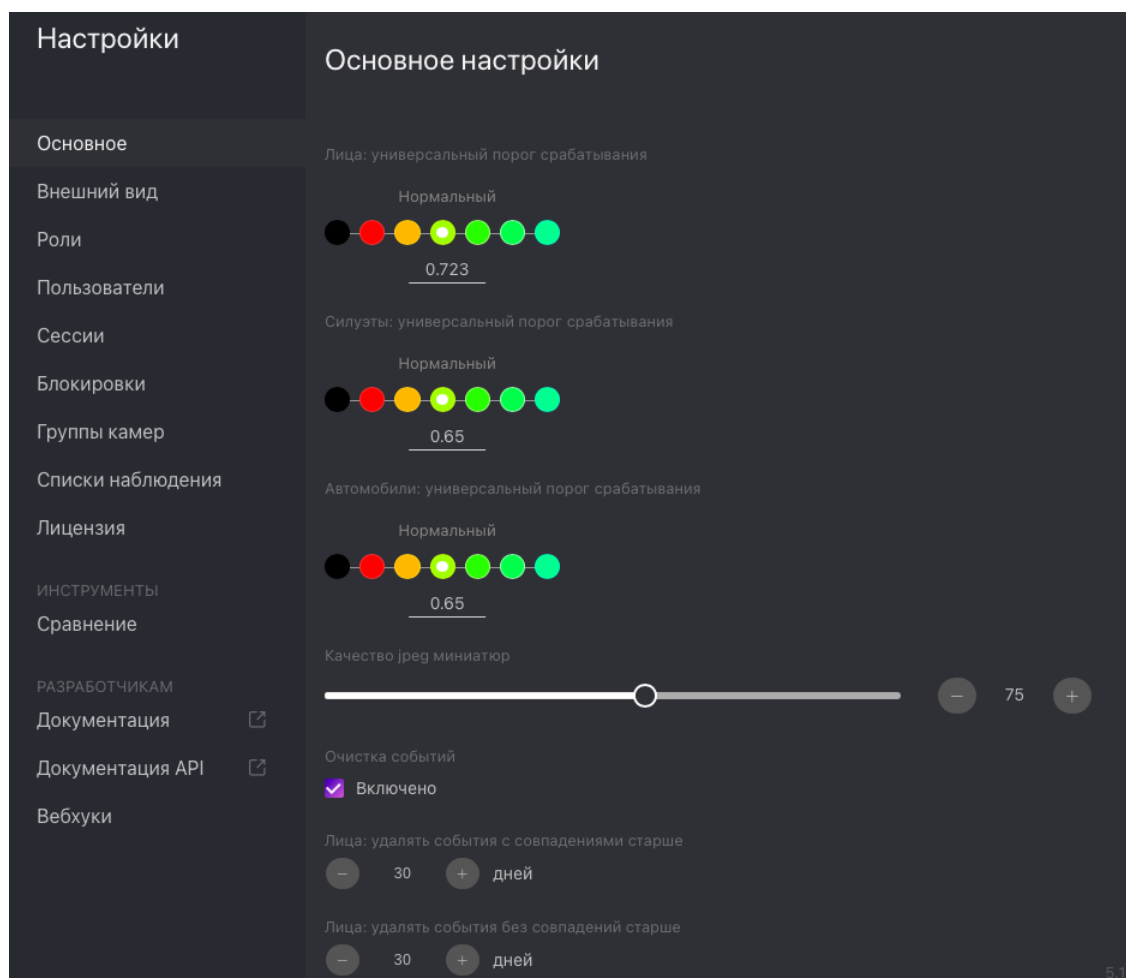
- Выберите продукт для установки: `NVIDIA CUDA drivers`.
- После завершения установки драйверов NVIDIA перезагрузите сервер.

2.5 Основные настройки

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

- универсальный порог срабатывания для распознавания лиц, силуэтов, автомобилей (в зависимости от включенных объектов)
- качество JPEG миниатюр
- расписание автоматического удаления старых событий и эпизодов

Для изменения основных настроек перейдите на вкладку *Настройки* и нажмите *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*

2.5.1 Универсальный порог срабатывания

FindFace Multi принимает решение о совпадении обнаруженного лица с лицом из картотеки (т. е. о совпадении лиц) на основании предустановленной пороговой степени схожести. По умолчанию установлено оптимальное пороговое значение. При необходимости вы можете изменить его.

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

Тот же принцип применяется к распознаванию силуэтов и автомобилей, если распознавание данных объектов активировано в системе. На основании предустановленного порога FindFace Multi принимает решение о принадлежности одному человеку обнаруженного силуэта и силуэта из картотеки. Аналогично, обнаруженный автомобиль будет считаться совпадающим с автомобилем из картотеки.

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).

2.5.2 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.

2.5.3 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. Вы также можете по отдельности удалять события с лицами, автомобилями и силуэтами.

2.6 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.

2.6.1 Управление пользователями

В этой главе:

- *Предустановленные роли*
- *Создание новой роли*
- *Главная и дополнительная роль пользователя*
- *Создание пользователя*
- *Деактивация или удаление пользователя*
- *Предоставление администратору прав на системные плагины*

Предустановленные роли

Для работы с FindFace Multi предусмотрены следующие предустановленные роли:

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

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

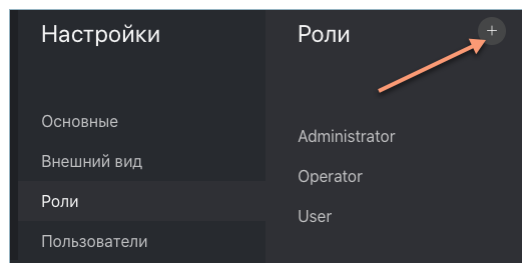
- **Оператор:** предоставляется полный доступ к функционалу FindFace Multi.
- **Пользователь:** обладает правами только на подтверждение событий и эпизодов, поиск объектов в базе событий, а также на изменение данных в своем профиле. Остальные функции доступны в режиме чтения.

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

Создание новой роли

Для создания новой роли выполните следующие действия:

1. Перейдите на вкладку *Настройки*. Выберите *Роли*.
2. Нажмите *+*.



3. На вкладке *Информация* задайте имя роли.

4. Нажмите *Сохранить*. Рядом с вкладкой *Информация* появятся дополнительные вкладки. На данных вкладках можно задать права на определенные списки наблюдения (вкладка *Списки наблюдения*) и группы камер (*Группы камер*), а также привилегии на работу с системными функциями и сущностями (*Разрешения*).

Примечание: Например, если вы устанавливаете **Нет** для определенной группы камер на вкладке *Группы камер*, пользователи с данной ролью не смогут работать с **этой** группой камер. Установка **Нет** для cameragroup на вкладке *Разрешения* не позволит пользователям просматривать и работать со **всеми** группами камер.

Примечание: Право на событие складывается из прав на соответствующую камеру и список наблюдения. Для просмотра событий, для которых не найдены совпадения с картотекой, требуются только права на камеру.

Полный список сущностей FindFace Multi:

- **area:** *область*
- **faceevent:** *событие распознавания лица*
- **faceobject:** *фотография лица в карточке*
- **carevent:** событие распознавания автомобиля
- **carobject:** фотография автомобиля в карточке
- **bodyevent:** событие распознавания силуэта
- **bodyobject:** фотография в полный рост в карточке
- **deviceblacklistrecord:** *список блокировок*
- **watchlist:** *список наблюдения*

- cameragroup: *группа камер*
- camera: *камера*
- uploadlist: список фотографий в *пакетной загрузке*
- upload: элемент (фото) в пакетной загрузке
- user: *пользователь*
- webhook: *вебхук*
- videoarchive: *идентификация объектов в видеофайлах*
- counter: *счетчики лиц и силуэтов*
- report: *отчеты*
- all_own_sessions: все *сессии* текущего пользователя на разных устройствах

Примечание: Если для данной сущности установлены соответствующие разрешения, пользователи смогут просматривать (**view**) и закрывать (**delete**) 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)*.

- humancard: *карточка человека*
- carcard: *карточка автомобиля*
- relation: *связь между карточками*
- humanepisode: *эпизод с человеком*
- carepisode: *эпизод с автомобилем*
- facecluster: *кластер лиц*
- bodycluster: *кластер силуэтов*
- carcluster: *кластер автомобилей*

Вы также можете включать и отключать права на следующий функционал:

- configure_ntls: конфигурация *сервера лицензий findface-ntls*
- batchupload_cards: *пакетная загрузка фотографий*
- view_runtimeetting: просмотр *основных настроек* FindFace Multi
- change_runtimeetting: изменение основных настроек FindFace Multi
- view_auditlog: просмотр и работа с *аудит логами*.

Информация Слiski наблюдения Группы камер <u>Разрешения</u>				
Выбрать все Отменить все				
Имя	Просмотр	Изменить	Добавить	Удалить
area	✓	✓	✓	✓
faceevent	✓	✓	✓	✓
faceobject	✓	✓	✓	✓
carevent	✓	✓	✓	✓
carobject	✓	✓	✓	✓
bodyevent	✓	✓	✓	✓
bodyobject	✓	✓	✓	✓
deviceblacklistrecord	✓	✓	✓	✓
watchlist	✓	✓	✓	✓
cameragroup	✓	✓	✓	✓
camera	✓	✓	✓	✓
uploadlist	✓	✓	✓	✓
upload	✓	✓	✓	✓
user	✓	✓	✓	✓
webhook	✓	✓	✓	✓
videoarchive	✓	✓	✓	✓
counter	✓	✓	✓	✓
metadictionary	✓	✓	✓	✓
notification	✓	✓	✓	✓
report	✓	✓	✓	✓
all_own_sessions	✓			✓
humancard	✓	✓	✓	✓
carcard	✓	✓	✓	✓
relation	✓	✓	✓	✓
humanepisode	✓	✓	✓	✓
carepisode	✓	✓	✓	✓
facecluster	✓	✓	✓	✓
bodycluster	✓	✓	✓	✓
carcluster	✓	✓	✓	✓
Имя	Активный			
configure_ntls				✓
batchupload_cards				✓
view_runtime-setting				✓
change_runtime-setting				✓
view_auditlog				✓

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 камерам внутри группы) и спискам наблюдения (и карточкам в этих списках), созданным пользователем с той же основной ролью, с учетом привилегий, определенных их дополнительными ролями.

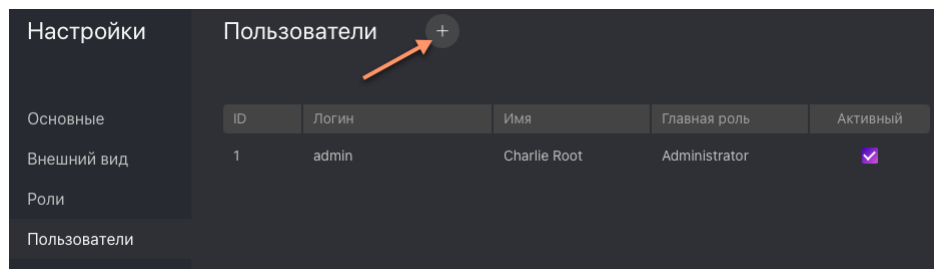
См.также:

Создание пользователя

Создание пользователя


Для создания нового пользователя выполните следующие действия:

1. Перейдите на вкладку *Настройки*. Выберите *Пользователи*.
2. Нажмите *+*.



3. Задайте имя пользователя, логин и пароль. При необходимости добавьте комментарий. Прикрепите фотографию пользователя.

Создать пользователя



* Имя

Фокс Малдер

* Логин

f.mulder@xfiles.org

* Пароль

••••••••

* Подтверждение пароля

••••••••

* Роли

User

Главная роль

☒ ☐

Добавить роль

Комментарий

Важно: Лицо на фотографии должно быть надлежащего качества, т. е. в близком к анфас положении. Расстояние между зрачками: 60 px. Поддерживаемые форматы: WEBP, JPG, BMP, PNG.

При несоответствии фотографии требованиям будет выведено сообщение с описанием ошибки.

Совет: Фото может использоваться для *биометрической аутентификации*.

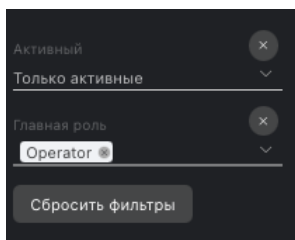
4. Из раскрывающегося меню *Roles* выберите одну или несколько пользовательских ролей. Назначьте одну из них основной.
5. Поставьте флажок *Активный*.
6. Нажмите *Создать*.

Деактивация или удаление пользователя

Для того чтобы деактивировать пользователя, снимите флажок *Активный* в списке пользователей (*Настройки* -> *Пользователи*).

Для удаления пользователя из FindFace Multi щелкните по его логину в списке. Нажмите *Удалить*.

Для фильтрации списка пользователей, используйте следующие критерии:



- *Активный*: статус пользователей
- *Главная роль*: одна или несколько главных ролей

Предоставление администратору прав на системные плагины

Пакет FindFace Multi включает в себя обширный набор системных плагинов, которые обеспечивают следующие функции:

- *интеграции с партнерами*
- авторизация с использованием криптографического сертификата (свяжитесь с менеджером, чтобы узнать подробности)

Примечание: Системные плагины включаются вручную через файл конфигурации `/etc/findface-security/config.py`.

По умолчанию Администратор не обладает правами на системные плагины. Для предоставления прав Администратору, выполните следующие действия:

1. Включите системный плагин в файле конфигурации `/etc/findface-security/config.py`, руководствуясь предоставленной нами пошаговой инструкцией.
2. Повторно выполните перенос основной архитектуры базы данных из FindFace Multi в PostgreSQL.

```
sudo findface-security migrate
```

3. Заново создайте группы пользователей в основной базе данных.

```
sudo findface-security create_groups
```

4. Перезапустите сервис `findface-security`.

```
sudo systemctl restart findface-security.service
```

2.6.2 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*.

Настройка аутентификации и мониторинга сессии

Для настройки аутентификации и мониторинга сессии пользователя выполните следующие действия:

1. Откройте файл конфигурации `/etc/findface-security/config.py`. Найдите разделы `FFSECURITY` и `FFSECURITY_AUTH_CONFIG`.

```
sudo vi /etc/findface-security/config.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 `findface-security`.

```
sudo systemctl restart findface-security.service
```

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 findface-security logout_all_users
```

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

2.6.3 Настройка шифрования данных

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

1. В директории с конфигурацией `nginx` создайте каталог для хранения информации о SSL-шифровании:

```
sudo mkdir /etc/nginx/ssl
```

2. Создайте ключ и сертификат SSL:

```
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. Настройте `nginx` для использования SSL. Откройте файл конфигурации `nginx /etc/nginx/sites-available/ffsecurity-nginx.conf`. Внесите в файл следующие изменения:

1. Добавьте новый раздел `server {...}`, содержащий правило замены URL:

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

(continues on next page)

(продолжение с предыдущей страницы)

```
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;
```

4. 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;
```

Пример файла конфигурации `/etc/nginx/sites-available/ffsecurity-nginx.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;
}

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;
```

(continues on next page)

(продолжение с предыдущей страницы)

```

autoindex off;

server_name _;

location = / {

    alias /usr/share/findface-security-ui/;
    try_files /index.html =404;
}
location /static/ {

}
location /uploads/ {
    add_header 'Access-Control-Allow-Origin' '*';
    add_header 'Access-Control-Allow-Methods' 'GET';
    add_header 'Access-Control-Allow-Headers' 'DNT,User-Agent,X-
↪Requested-With,If-Modified-Since,Cache-Control,Content-Type,Range,
↪Authorization';
    add_header 'Access-Control-Expose-Headers' 'Content-Length,
↪Content-Range';
    add_header 'Access-Control-Max-Age' 2592000;
}
location /ui-static/ {
    alias /usr/share/findface-security-ui/ui-static/;
}
location /doc/ {
    alias /opt/findface-security/doc/;
}
location ~ /videos/(?<video_id>[0-9]+)/upload/(.*)$ {
    if ($request_method = 'OPTIONS') {
        add_header 'Content-Type' 'text/plain; charset=utf-8';
        add_header 'Content-Length' 0;
        return 204;
    }
    set $auth_request_uri "http://ffsecurity/videos/$video_id/auth-
↪upload/";
    auth_request /video-upload-auth/;

    alias "/var/lib/findface-security/uploads/videos/$video_id.bin";
    client_max_body_size 15g;

    dav_access user:rw group:rw all:rw;
    dav_methods PUT;

    create_full_put_path on;
    autoindex off;
    autoindex_exact_size off;
    autoindex_localtime on;
    charset utf-8;

    add_header 'Access-Control-Allow-Origin' '*';

```

(continues on next page)

(продолжение с предыдущей страницы)

```

    add_header 'Access-Control-Allow-Methods' 'PUT, OPTIONS';
    add_header 'Access-Control-Allow-Headers' 'authorization';
}
location = /video-upload-auth/ {
    internal;
    client_max_body_size 15g;
    proxy_set_header Content-Length "";
    proxy_set_header Host $http_host;
    proxy_set_header X-Forwarded-For $remote_addr;
    proxy_set_header X-Forwarded-Proto $scheme;
    proxy_pass_request_body off;
    proxy_pass $auth_request_uri;
}

location / {
    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_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)/([0-9]+)/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;
    }
}
}

```

4. Перезапустите nginx.

```
sudo systemctl restart nginx.service
```

5. Внесите изменения в файл конфигурации /etc/findface-security/config.py. В параметрах EXTERNAL_ADDRESS и ROUTER_URL измените приставку http:// на https://.

```

sudo vi /etc/findface-security/config.py

...
EXTERNAL_ADDRESS="https://my-example-domain.com"
...
ROUTER_URL="https://IP_address"

```

6. Перезапустите `findface-security`.

```
sudo systemctl restart findface-security
```

7. Если есть запущенные процессы `findface-video-worker`, нужно либо пересоздать камеры в веб-интерфейсе, либо изменить значение параметра `router_url` в job-заданиях, заменив приставку `http://` на `https://`. Это можно сделать с помощью команды, аналогичной следующей:

```
curl -s localhost:18810/jobs | jq -r '.[]["id"]' | xargs -I {} curl -X PATCH -d '{
  ↪ "router_url": "https://my-example-domain.com/video-detector/frame"}' http://
  ↪ localhost:18810/job/{}'
```

2.6.4 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 автомобилей. Для этого перейдите *Настройки* → *Роли* → нужная роль → *Разрешения* 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:

Настройка мониторинга объектов и картотеки.

Для включения защиты картотеки выполните следующие действия:

1. Откройте файл конфигурации `/etc/findface-security/config.py`.

```
sudo vi /etc/findface-security/config.py
```

2. Раскомментируйте параметр `OVERPROTECT_MEDIA` и установите его в значение `True`.

```
...

'OVERPROTECT_MEDIA': False,
```

3. Откройте файл конфигурации nginx `/etc/nginx/sites-available/ffsecurity-nginx.conf`. Раскомментируйте параметр `internal` в разделе `location /uploads`.

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

4. Перезапустите `findface-security` и `nginx`.

```
sudo systemctl restart findface-security.service
sudo systemctl restart nginx.service
```

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 findface-security logout_all_users
```

2.6.5 Отключение ACL

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

Выполните следующие действия:

1. Откройте файл конфигурации `/etc/findface-security/config.py`.

```
sudo vi /etc/findface-security/config.py
```

2. Установите `ENABLE_ACL = False`.

```
...
ENABLE_ACL = False
```

3. Перезапустите `findface-security`.

```
sudo systemctl restart findface-security.service
```

2.6.6 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*.

Журнал действий						
<div> <div>«</div> <div><</div> <div>Страница 1</div> <div>></div> <div>»</div> </div>						
Пользователь	IP	ID устройства	Действие	Объект	ID объекта	Время
admin	172.20.78.22	585607ef-3551-4eb2-a54f-1bca5568617f	Редактировать	Событие с лицом	4377324716178297828	2021-09-06 15:06:53
admin	172.20.177.84	31f02282-38c5-4c6d-9dd0-94c525b3b5ec	Авторизация	Пользователь	1	2021-09-04 02:04:39
admin	172.20.78.30	76f2705f-264c-45c2-9a94-502be31d51d6	Редактировать	Камера	1	2021-09-03 22:42:20
admin	172.20.78.30	76f2705f-264c-45c2-9a94-502be31d51d6	Редактировать	Камера	3	2021-09-03 2
admin	172.20.78.34	51c841fe-2daf-4a9e-a6d7-ba06df6f2a78	Редактировать	Счетчик	1	2021-09-03 2
admin	172.20.78.34	51c841fe-2daf-4a9e-a6d7-ba06df6f2a78	Редактировать	Счетчик	2	2021-09-03 2
admin	172.20.78.82	b6eaa18-8dcd-4390-8292-af91e87aa659	Авторизация	Пользователь	1	2021-09-03 20:57:43
admin	172.20.78.82	b6eaa18-8dcd-4390-8292-af91e87aa659	Авторизация	Пользователь	1	2021-09-03 20:57:42
admin	172.20.178.10	65fdd052-8abe-4852-bf06-3ee89e0d9266	Редактировать	Счетчик	3	2021-09-03 2
admin	172.20.178.10	65fdd052-8abe-4852-bf06-3ee89e0d9266	Редактировать	Счетчик	3	2021-09-03 20:34:47
admin	172.20.178.10	65fdd052-8abe-4852-bf06-3ee89e0d9266	Создать	Счетчик	3	2021-09-03 20:34:30
admin	172.20.178.10	65fdd052-8abe-4852-bf06-3ee89e0d9266	Создать	Счетчик	2	2021-09-03 20:33:50
admin	172.20.78.58	4f282c24-1408-404a-8afe-cc4f55d04bcc	Авторизация	Пользователь	1	2021-09-03 1
admin	172.20.78.30		Авторизация	Пользователь	1	2021-09-03 1
admin	172.20.78.22	585607ef-3551-4eb2-a54f-1bca5568617f	Создать	Счетчик	1	2021-09-03 1

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 или камера
- идентификатор объекта
- подробности в зависимости от типа действия
- временная метка

Используйте панель фильтров справа, чтобы задать условия поиска в журнале.

2.6.7 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 (мобильное приложение или веб-интерфейс), 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 на вкладку *Настройки* -> *Сессии*.

Настройки

НазадСессииСессииБлокировки

Основное

БлокироватьЗаккрыть

Страница 1

Внешний вид

<input type="checkbox"/>	UUID	Логин	Информация об устройстве	Устройство	IP	Статус	Последнее соединение
<input type="checkbox"/>	d01914cb-3cb6-4646-b06a-6a6952679f4a	admin	{ "user_agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/14.1.1 Safari/605.1.15" }	web	172.20.78.58	онлайн	2021-10-05 15:34:22
<input type="checkbox"/>	4af2b002-6c34-4ecd-b9c7-7c2f2ff541cb	admin	{ "user_agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/94.0.4606.61 Safari/537.36" }	web	172.20.78.66	онлайн	2021-10-05 15:34:21

Роли

Пользователи

Сессии

Блокировки

Группы камер

Списки наблюдения

Лицензия

ИНСТРУМЕНТЫ

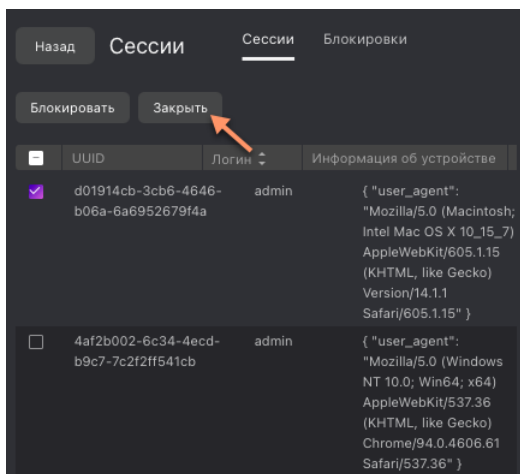
Сравнение

Каждая запись сессии содержит следующие данные:

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

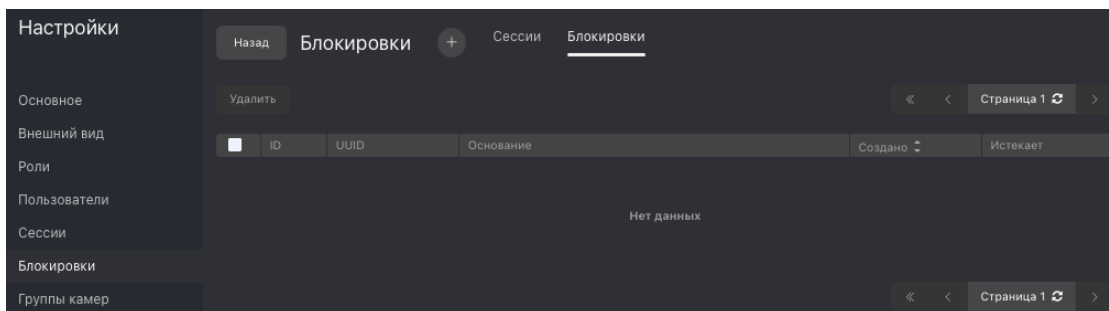
Используйте панель фильтров справа, чтобы задать условия поиска в списке сессий.

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



Блокировка устройства

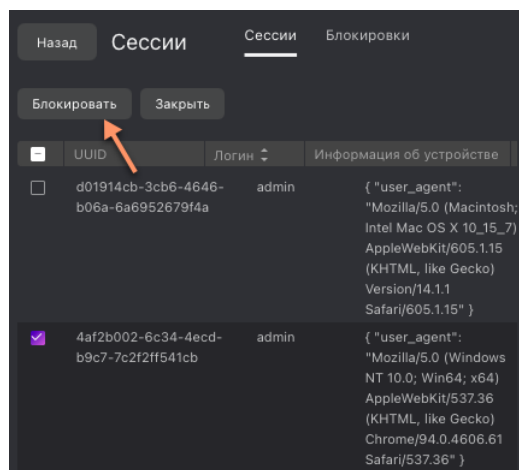
Список заблокированных устройств доступен на вкладке *Блокировки (Настройки)*.



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

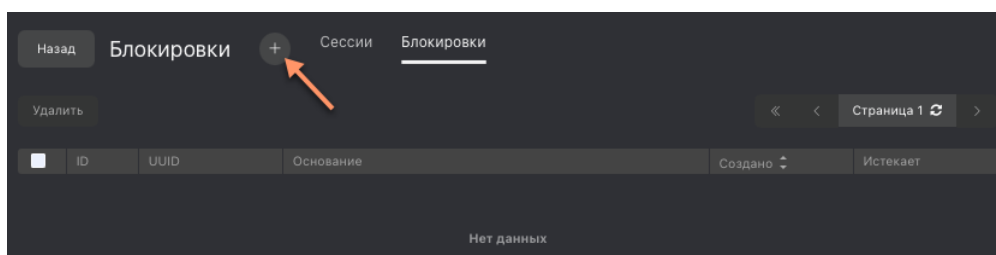
Для блокировки устройства на вкладке *Сессии* выполните следующие действия:

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



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

Блокировка устройства на вкладке *Блокировки* выполняется аналогично:



1. Нажмите *+*.
2. Вручную введите UUID устройства.
3. Укажите основание и срок действия блокировки.
4. Нажмите *Сохранить*.

2.6.8 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:

1. Откройте файл конфигурации `/etc/findface-security/config.py`.

```
sudo vi /etc/findface-security/config.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 сервис `findface-security`.

```
sudo systemctl restart findface-security.service
```

2.7 Настройка расширенного функционала

2.7.1 Распознавание атрибутов лица

FindFace Multi позволяет автоматически распознавать в реальном времени такие атрибуты лица, как пол, возраст, эмоции, очки, борода и медицинская маска. Данный функционал доступен на видеопотоке объектов с ускорением на GPU и CPU.

Для включения автоматического распознавания атрибутов лиц выполните следующие действия:

1. Откройте файл конфигурации `/etc/findface-extraction-api.ini`.

```
sudo vi /etc/findface-extraction-api.ini
```

2. В секции `extractors` укажите соответствующие модели, 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`. 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.

```
extractors:
  face_age: faceattr/age.v2.gpu.fnk
  face_beard: faceattr/beard.v0.gpu.fnk
  face_emotions: faceattr/emotions.v1.gpu.fnk
  face_gender: faceattr/gender.v2.gpu.fnk
  face_glasses3: faceattr/glasses3.v0.gpu.fnk
  face_medmask3: faceattr/medmask3.v2.gpu.fnk
```

Доступны следующие модели:

Атрибут лица	Ускорение	Настройка
возраст	CPU	face_age: faceattr/age.v2.cpu.fnk
	GPU	face_age: faceattr/age.v2.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
борода	CPU	face_beard: faceattr/beard.v0.cpu.fnk
	GPU	face_beard: faceattr/beard.v0.gpu.fnk
медицинская маска	CPU	face_medmask3: faceattr/medmask3.v2.cpu.fnk
	GPU	face_medmask3: faceattr/medmask3.v2.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_emotions: ""
  face_gender: ""
  face_glasses3: ""
  face_medmask3: ""
```

Примечание: Вы можете найти модели для распознавания атрибутов лица в каталоге `/usr/share/findface-data/models/faceattr/`.

```
ls /usr/share/findface-data/models/faceattr/
age.v2.cpu.fnk age.v2.gpu.fnk beard.v0.cpu.fnk beard.v0.gpu.fnk emotions.v1.cpu.
↪fnk emotions.v1.gpu.fnk gender.v2.cpu.fnk gender.v2.gpu.fnk glasses3.v0.cpu.
↪fnk glasses3.v0.gpu.fnk medmask3.v2.cpu.fnk medmask3.v2.gpu.fnk liveness.colombo.
↪cpu.fnk liveness.colombo.gpu.fnk liveness.pacs.v0.cpu.fnk liveness.pacs.v0.gpu.
↪fnk quality.v1.cpu.fnk quality.v1.gpu.fnk
```

3. Перезапустите `findface-extraction-api`.

```
sudo systemctl restart findface-extraction-api
```

4. Для того чтобы отображать результаты распознавания атрибутов лиц в списке событий, откройте файл конфигурации `/etc/findface-security/config.py`.

```
sudo vi /etc/findface-security/config.py
```

5. Укажите включенные модели в следующей строке раздела `FFSECURITY`:

```
FFSECURITY = {
    ...
```

(continues on next page)

(продолжение с предыдущей страницы)

```
'FACE_EVENTS_FEATURES': ['gender', 'age', 'emotions', 'beard', 'glasses',
↪ 'medmask'],
...
}
```

6. Перезапустите `findface-security`.

```
sudo systemctl restart findface-security
```

2.7.2 Распознавание автомобилей и их атрибутов

FindFace Multi позволяет распознавать отдельные автомобили и их атрибуты.

Атрибуты автомобилей следующие:

- номерной знак (для отдельных стран),
- цвет,
- марка,
- модель,
- тип кузова,
- принадлежность к специальному транспорту (полиция, скорая помощь, спасательные службы, такси).

Важно: Распознавание отдельных автомобилей является экспериментальной функцией. Поэтому для того чтобы улучшить качество распознавания, мы настоятельно рекомендуем вам включить дополнительный анализ атрибутов. В этом случае система будет сравнивать не только векторы признаков двух автомобилей, но также их атрибуты, такие как цвет, тип кузова, марка, модель, принадлежность к специальному транспорту. Вывод о совпадении автомобилей будет делаться только в том случае, если совпадают как векторы признаков, так и атрибуты автомобилей.

Подробное описание того, как включить дополнительный анализ атрибутов, вы найдете в пошаговой инструкции ниже.

Для включения распознавания автомобилей и их атрибутов выполните следующие действия:

1. В файле конфигурации `/etc/findface-extraction-api.ini` укажите модели нейронных сетей для распознавания объектов типа автомобиль и их атрибутов. Выполните следующие действия:

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 `/etc/findface-extraction-api.ini`.

```
sudo vi /etc/findface-extraction-api.ini
```

2. Укажите модель детектора автомобилей в разделе `detectors -> models`, вставив следующий код:

GPU

```

detectors:
    ...
    models:
        ...
        efreitor:
            aliases:
                - car
            model: cadet/efreitor.gpu.fnk
            options:
                min_object_size: 32
                resolutions: [256x256, 384x384, 512x512, 768x768, 1024x1024, 1536x1536,
↵2048x2048]
        ...

```

CPU

```

detectors:
    ...
    models:
        ...
        efreitor:
            aliases:
                - car
            model: cadet/efreitor.cpu.fnk
            options:
                min_object_size: 32
                resolutions: [256x256, 384x384, 512x512, 768x768, 1024x1024, 1536x1536,
↵2048x2048]
        ...

```

3. Укажите модели для экстракции векторов признаков в разделе **extractors** -> **models**, в зависимости от необходимых экстракторов:

GPU

```

extractors:
    ...
    models:
        car_color: ''
        car_description: carattr/description.v0.gpu.fnk
        car_emben: carrec/alonso.gpu.fnk
        car_license_plate: carattr/carattr.license_plate.v4.gpu.fnk
        car_license_plate_quality: carattr/carattr.license_plate_quality.v0.gpu.fnk
        car_make: ''

```

(continues on next page)

(продолжение с предыдущей страницы)

```
car_quality: carattr/carattr.quality.v0.gpu.fnk
car_special_types: carattr/carattr.special_types.v0.gpu.fnk
```

CPU

```
extractors:
...
models:
  car_color: ''
  car_description: carattr/description.v0.cpu.fnk
  car_emben: carrec/alonso.cpu.fnk
  car_license_plate: carattr/carattr.license_plate.v4.cpu.fnk
  car_license_plate_quality: carattr/carattr.license_plate_quality.v0.cpu.fnk
  car_make: ''
  car_quality: carattr/carattr.quality.v0.cpu.fnk
  car_special_types: carattr/carattr.special_types.v0.cpu.fnk
```

Доступны следующие экстракторы:

Экстрактор	Настройка
отдельный объект типа автомобиля	car_emben: carrec/alonso.cpu.fnk
	car_emben: carrec/alonso.gpu.fnk
номерной знак	car_license_plate: carattr/carattr.license_plate.v4.cpu.fnk car_license_plate_quality: carattr/carattr.license_plate_quality.v0.cpu.fnk
	car_license_plate: carattr/carattr.license_plate.v4.gpu.fnk car_license_plate_quality: carattr/carattr.license_plate_quality.v0.gpu.fnk
набор атрибутов: марка / цвет / модель / тип кузова	car_description: carattr/description.v0.cpu.fnk
	car_description: carattr/description.v0.gpu.fnk
качество изображения автомобиля	car_quality: carattr/carattr.quality.v0.cpu.fnk
	car_quality: carattr/carattr.quality.v0.gpu.fnk
спецтранспорт	car_special_types: carattr/carattr.special_types.v0.cpu.fnk
	car_special_types: carattr/carattr.special_types.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_color: ""
  car_description: ""
  car_emben: ""
```

(continues on next page)

(продолжение с предыдущей страницы)

```

car_license_plate: ""
car_license_plate_quality: ""
car_make: ""
car_quality: ""
car_special_types: ""

```

4. Укажите нормализаторы, необходимые для работы экстракторов, заданных на предыдущем шаге. Например, если вам требуется настроить распознавание автомобильных номеров, укажите нормализатор `carlicplate`.

Нормализатор	Модель нормализатора	Используется для экстракторов
carlicplate	carnorm/anaferon.v3.gpu.fnk carnorm/anaferon.v3.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_types

GPU

```

normalizers:
    ...

models:
    carlicplate:
        model: carnorm/anaferon.v3.gpu.fnk
    ...
    cropbbox:
        model: facenorm/cropbbox.v2.gpu.fnk

```

CPU

```

normalizers:
    ...

models:
    carlicplate:
        model: carnorm/anaferon.v3.cpu.fnk
    ...
    cropbbox:
        model: facenorm/cropbbox.v2.cpu.fnk

```

5. Убедитесь, что раздел `objects -> car` содержит `quality_attribute: car_quality`:

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
```

6. Перезапустите findface-extraction-api.

```
sudo systemctl restart findface-extraction-api
```

2. Внесите изменения в файл конфигурации /etc/findface-video-worker-gpu.ini (/etc/findface-video-worker-cpu.ini). В разделе car укажите модели нейронных сетей по аналогии с примером ниже. Перезапустите findface-video-worker-gpu (findface-video-worker-cpu).

GPU

```
sudo vi /etc/findface-video-worker-gpu.ini

#-----
[car]
#-----
## detector param
## type:number env:CFG_CAR_MIN_SIZE longopt:--car-min-size
min_size = 60

## path to car detector
## type:string env:CFG_CAR_DETECTOR longopt:--car-detector
detector = /usr/share/findface-data/models/cadet/efreitor.gpu.fnk

## path to normalizer (usually crop2x)
## type:string env:CFG_CAR_NORM longopt:--car-norm
norm = /usr/share/findface-data/models/facenorm/cropbbox.v2.gpu.fnk

## path to car quality extractor
## type:string env:CFG_CAR_QUALITY longopt:--car-quality
quality = /usr/share/findface-data/models/carattr/carattr.quality.v0.gpu.fnk

## path to car quality normalizer
```

(continues on next page)

(продолжение с предыдущей страницы)

```
## type:string env:CFG_CAR_NORM_QUALITY longopt:--car-norm-quality
norm_quality = /usr/share/findface-data/models/facenorm/cropbbox.v2.gpu.fnk

## path to car track features extractor
## type:string env:CFG_CAR_TRACK_FEATURES longopt:--car-track-features
track_features =

## path to car track features normalizer
## type:string env:CFG_CAR_TRACK_FEATURES_NORM longopt:--car-track-features-norm
track_features_norm =
```

```
sudo systemctl restart findface-video-worker-gpu.service
```

CPU

```
sudo vi /etc/findface-video-worker-cpu.ini

#-----
[car]
#-----
## detector param
## type:number env:CFG_CAR_MIN_SIZE longopt:--car-min-size
min_size = 60

## path to car detector
## type:string env:CFG_CAR_DETECTOR longopt:--car-detector
detector = /usr/share/findface-data/models/cadet/efreitor.cpu.fnk

## path to normalizer (usually crop2x)
## type:string env:CFG_CAR_NORM longopt:--car-norm
norm = /usr/share/findface-data/models/facenorm/cropbbox.v2.cpu.fnk

## path to car quality extractor
## type:string env:CFG_CAR_QUALITY longopt:--car-quality
quality = /usr/share/findface-data/models/carattr/carattr.quality.v0.cpu.fnk

## path to car quality normalizer
## type:string env:CFG_CAR_NORM_QUALITY longopt:--car-norm-quality
norm_quality = /usr/share/findface-data/models/facenorm/cropbbox.v2.cpu.fnk
```

```
sudo systemctl restart findface-video-worker-cpu.service
```

- Откройте файл конфигурации `/etc/findface-video-manager.conf` и убедитесь, что он содержит раздел `car` в `detectors`, аналогичный приведенному ниже.

Совет: В качестве эталонного значения параметра `filter_min_quality` можно взять значение параметра `MINIMUM_CAR_QUALITY` из файла конфигурации `/etc/findface-security/config.py`.

```

sudo vi /etc/findface-video-manager.conf

detectors:
...
car:
  filter_min_quality: 0.65
  filter_min_size: 1
  filter_max_size: 8192
  roi: ""
  fullframe_crop_rot: false
  fullframe_use_png: false
  jpeg_quality: 95
  overall_only: false
  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

```

4. Включите распознавание автомобилей и атрибутов автомобилей в файле конфигурации `/etc/findface-security/config.py`. Выполните следующие действия:

1. В разделе FFSECURITY установите 'ENABLE_CARS': True.

```

sudo vi /etc/findface-security/config.py

FFSECURITY = {
...

    # optional objects to detect
    'ENABLE_CARS': True,
...

```

2. В том же разделе укажите атрибуты автомобилей, которые требуется отображать в событиях распознавания.

```

# available features are: description, license_plate, special_vehicle_type
'CAR_EVENTS_FEATURES': ['description', 'license_plate', 'special_vehicle_type'],

```

3. Для того чтобы улучшить качество распознавания отдельных автомобилей, мы настоятельно рекомендуем вам включить дополнительный анализ атрибутов. В этом случае система будет сравнивать не только векторы признаков двух автомобилей, но и атрибуты автомобилей. Вывод о совпадении автомобилей будет делаться только в том случае, если совпадают как векторы признаков, так и атрибуты.

Для дополнительного анализа вы можете использовать следующие атрибуты:

- `color`: цвет автомобиля,
- `body`: тип кузова,
- `make`: марка,
- `model`: модель,
- `special_vehicle_type`: принадлежность к специальному транспорту.

Для того чтобы включить дополнительный анализ атрибутов, установите `True` в разделе `FFSECURITY -> EXTRA_CAR_MATCHING` для атрибутов, которые вы хотите сравнивать.

```
FFSECURITY = {
    # use additional features for extra confidence when matching cars by emben
    'EXTRA_CAR_MATCHING': {
        'color': {'enabled': False, 'min_confidence': 0},
        'body': {'enabled': False, 'min_confidence': 0},
        'make': {'enabled': False, 'min_confidence': 0},
        'model': {'enabled': False, 'min_confidence': 0},
        'special_vehicle_type': {'enabled': False, 'min_confidence': 0}
    },
}
```

Важно: Для того чтобы анализ атрибутов работал, в файлах конфигурации `/etc/findface-extraction-api.ini` и `/etc/findface-security/config.py` должна быть включена модель `description` (см. выше).

Примечание: Включение дополнительного анализа атрибутов уменьшает количество ложноположительных срабатываний, однако могут быть пропущены также некоторые реальные совпадения.

Предупреждение: Не меняйте значения по умолчанию, установленные для `min_confidence`, без предварительной консультации с нашими техническими специалистами (support@ntechlab.com).

4. Перезапустите сервис `findface-security`.

```
sudo systemctl restart findface-security.service
```

2.7.3 Распознавание силуэтов и их атрибутов

FindFace Multi позволяет распознавать отдельные человеческие силуэты и атрибуты силуэтов.

Атрибуты силуэтов следующие:

- 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)

Для включения распознавания силуэтов и их атрибутов выполните следующие действия:

1. В файле конфигурации `/etc/findface-extraction-api.ini` укажите модели нейронных сетей для распознавания объектов типа силуэт и их атрибутов. Выполните следующие действия:

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 `/etc/findface-extraction-api.ini`.

```
sudo vi /etc/findface-extraction-api.ini
```

2. Укажите модель детектора силуэтов в разделе `detectors -> models`, вставив следующий код:

GPU

```
detectors:
    ...
    models:
        ...
        glenn:
            aliases:
                - body
                - silhouette
            model: pedet/glenn_005.gpu.fnk
            options:
                min_object_size: 32
                resolutions: [256x256, 384x384, 512x512, 768x768, 1024x1024, 1536x1536,
↪ 2048x2048]
        ...
```

CPU

```
detectors:
    ...
    models:
        ...
        glenn:
            aliases:
                - body
                - silhouette
            model: pedet/glenn_005.cpu.fnk
            options:
                min_object_size: 32
                resolutions: [256x256, 384x384, 512x512, 768x768, 1024x1024, 1536x1536, ↵
↵2048x2048]
        ...
```

3. Укажите модели для экстракции векторов признаков в разделе **extractors** -> **models**, в зависимости от необходимых экстракторов:

GPU

```
extractors:
    ...
    models:
        body_clothes: pedattr/pedattr.clothes_type.v0.gpu.fnk
        body_color: pedattr/pedattr.color.v1.gpu.fnk
        body_emben: pedrec/andariel.gpu.fnk
        body_quality: pedattr/pedattr.quality.v0.gpu.fnk
```

CPU

```
extractors:
    ...
    models:
        body_clothes: pedattr/pedattr.clothes_type.v0.cpu.fnk
        body_color: pedattr/pedattr.color.v1.cpu.fnk
        body_emben: pedrec/andariel.cpu.fnk
        body_quality: pedattr/pedattr.quality.v0.cpu.fnk
```

Доступны следующие экстракторы:

Тип распознавания	Настройка
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
отдельный объект типа силуэт	body_emben: pedrec/andariel.gpu.fnk
	body_emben: pedrec/andariel.cpu.fnk
качество изображения силуэта	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.

```
extractors:
...
models:
  body_clothes: ''
  body_color: ''
  body_emben: ''
  body_quality: ''
```

- Убедитесь, что раздел **normalizers** содержит модель для нормализатора **cropbbox**, как показано в примере ниже. Данный нормализатор требуется для работы экстракторов, указанных на предыдущем шаге.

GPU

```
normalizers:
...
models:
...
  cropbbox:
    model: facenorm/cropbbox.v2.gpu.fnk
```

CPU

```
normalizers:
...
models:
...
  cropbbox:
    model: facenorm/cropbbox.v2.cpu.fnk
```

- Убедитесь, что раздел **objects** -> **body** содержит **quality_attribute: body_quality:**

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
```

6. Перезапустите findface-extraction-api.

```
sudo systemctl restart findface-extraction-api
```

2. Внесите изменения в файл конфигурации /etc/findface-video-worker-gpu.ini (/etc/findface-video-worker-cpu.ini). В разделе body укажите модели нейронных сетей по аналогии с примером ниже. Перезапустите findface-video-worker-gpu (findface-video-worker-cpu).

GPU

```
sudo vi /etc/findface-video-worker-gpu.ini

#-----
[body]
#-----
## detector param
## type:number env:CFG_BODY_MIN_SIZE longopt:--body-min-size
min_size = 60

## path to body detector
## type:string env:CFG_BODY_DETECTOR longopt:--body-detector
detector = /usr/share/findface-data/models/pedet/glenny_005_fast.gpu.fnk

## path to normalizer (usually crop2x)
## type:string env:CFG_BODY_NORM longopt:--body-norm
norm = /usr/share/findface-data/models/facenorm/cropbbox.v2.gpu.fnk

## path to body quality extractor
## type:string env:CFG_BODY_QUALITY longopt:--body-quality
quality = /usr/share/findface-data/models/pedattr/pedattr.quality.v0.gpu.fnk

## path to body quality normalizer
## type:string env:CFG_BODY_NORM_QUALITY longopt:--body-norm-quality
norm_quality = /usr/share/findface-data/models/facenorm/cropbbox.v2.gpu.fnk
```

```
sudo systemctl restart findface-video-worker-gpu.service
```

CPU

```
sudo vi /etc/findface-video-worker-cpu.ini

#-----
[body]
#-----
## detector param
## type:number env:CFG_BODY_MIN_SIZE longopt:--body-min-size
min_size = 60

## path to body detector
## type:string env:CFG_BODY_DETECTOR longopt:--body-detector
detector = /usr/share/findface-data/models/pedet/glenny_005_fast.cpu.fnk

## path to normalizer (usually crop2x)
## type:string env:CFG_BODY_NORM longopt:--body-norm
norm = /usr/share/findface-data/models/facenorm/cropbbox.v2.cpu.fnk

## path to body quality extractor
## type:string env:CFG_BODY_QUALITY longopt:--body-quality
quality = /usr/share/findface-data/models/pedattr/pedattr.quality.v0.cpu.fnk

## path to body quality normalizer
## type:string env:CFG_BODY_NORM_QUALITY longopt:--body-norm-quality
norm_quality = /usr/share/findface-data/models/facenorm/cropbbox.v2.cpu.fnk

## path to body track features extractor
## type:string env:CFG_BODY_TRACK_FEATURES longopt:--body-track-features
track_features =

## path to body track features normalizer
## type:string env:CFG_BODY_TRACK_FEATURES_NORM longopt:--body-track-features-norm
track_features_norm =
```

```
sudo systemctl restart findface-video-worker-cpu.service
```

- Откройте файл конфигурации `/etc/findface-video-manager.conf` и убедитесь, что он содержит раздел `body` в `detectors`, аналогичный приведенному ниже.

Совет: В качестве эталонного значения параметра `filter_min_quality` можно взять значение параметра `MINIMUM_BODY_QUALITY` из файла конфигурации `/etc/findface-security/config.py`.

```
sudo vi /etc/findface-video-manager.conf

detectors:
    ...
```

(continues on next page)

(продолжение с предыдущей страницы)

```
body:
  filter_min_quality: 0.65
  filter_min_size: 1
  filter_max_size: 8192
  roi: ""
  fullframe_crop_rot: false
  fullframe_use_png: false
  jpeg_quality: 95
  overall_only: false
  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
```

4. Включите распознавание силуэтов и атрибутов силуэтов в файле конфигурации `/etc/findface-security/config.py`. Выполните следующие действия:

1. В разделе FFSECURITY установите `'ENABLE_BODIES: True'`.

```
sudo vi /etc/findface-security/config.py

FFSECURITY = {
    ...

    # optional objects to detect
    'ENABLE_BODIES': True,
    ...
```

2. В том же разделе укажите атрибуты силуэтов, которые требуется отображать в событиях распознавания.

```
# available features are: color, clothes
'BODY_EVENTS_FEATURES': ['color', 'clothes'],
```

3. Перезапустите сервис `findface-security`.

```
sudo systemctl restart findface-security.service
```


2.7.4 Распознавание живых лиц в реальном времени (Liveness)

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

Для обнаружения фальшивых лиц и предотвращения фото-атак используйте интегрированную антиспуфинговую систему, отличающую живые лица от их изображений. Алгоритм анализирует несколько последовательных кадров, регистрируя изменения в мимике и текстуре кожи, и благодаря этому определяет, является ли лицо перед камерой живым или фальшивым. Это исключает возможность мошенничества с использованием изображения лица на бумаге или экране мобильного устройства.

Liveness-детектор оценивает живость лица с определенным уровнем достоверности и возвращает оценку достоверности вместе с бинарным результатом **Живой человек/изображение**, в зависимости от установленного порога достоверности.

В этом разделе:

- Включение *Liveness-детектора*
- Настройка порога *Liveness*
- Информация о живом лице в веб-интерфейсе

Включение Liveness-детектора

Для включения Liveness-детектора выполните следующие действия:

1. Откройте файл конфигурации `/etc/findface-video-worker-gpu.ini` (`/etc/findface-video-worker-cpu.ini`). В разделе `liveness` укажите путь к модели нейронной сети (`fnk`), которая будет использоваться в `liveness-детекторе`.

```
sudo vi /etc/findface-video-worker-gpu.ini

#-----
[liveness]
#-----
## path to liveness fnk
## type:string env:CFG_LIVENESS_FNK longopt:--liveness-fnk
fnk = /usr/share/findface-data/models/faceattr/liveness.pacs.v0.gpu.fnk
```

```
sudo vi /etc/findface-video-worker-cpu.ini

#-----
[liveness]
#-----
## path to liveness fnk
## type:string env:CFG_LIVENESS_FNK longopt:--liveness-fnk
fnk = /usr/share/findface-data/models/faceattr/liveness.pacs.v0.cpu.fnk
```

2. Перезапустите `findface-video-worker`.

```
sudo systemctl restart findface-video-worker-gpu
sudo systemctl restart findface-video-worker-cpu
```

Настройка порога Liveness

При необходимости вы можете настроить пороговое значение Liveness в файле конфигурации `/etc/findface-security/config.py`. Liveness-детектор оценивает “живость” лица с определенной достоверностью. В зависимости от порогового значения достоверности, он возвращает бинарный результат Живой человек или Изображение.

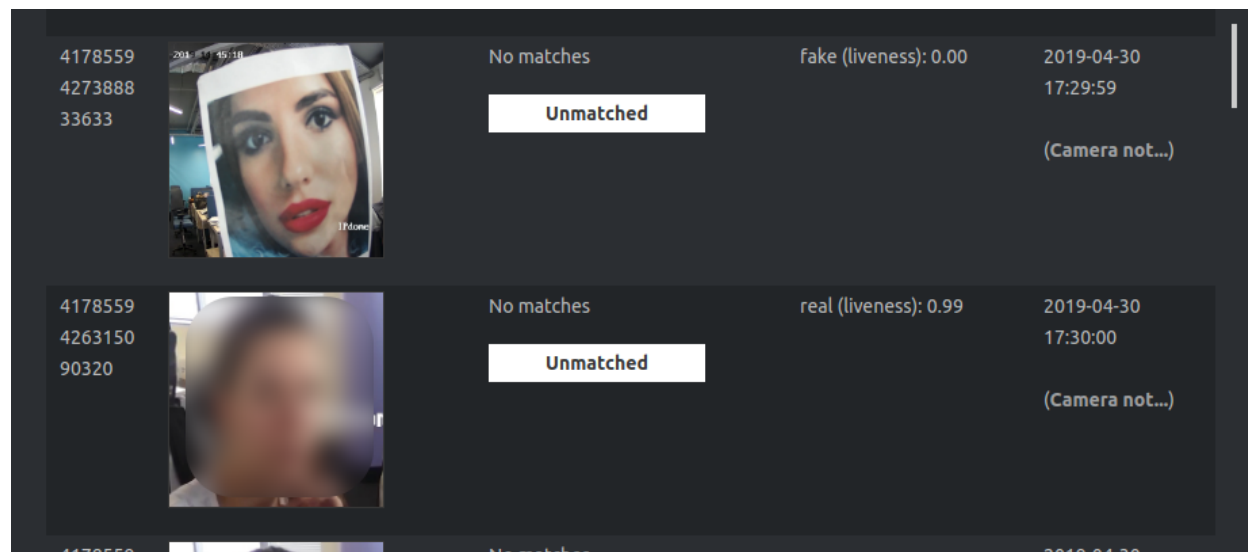
Примечание: Значение по умолчанию является оптимальным. Перед изменением порога проконсультируйтесь у наших специалистов по адресу support@ntechlab.com.

```
sudo vi /etc/findface-security/config.py

'LIVENESS_THRESHOLD': 0.85,
```

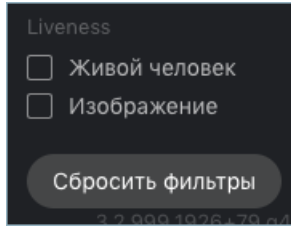
Информация о живом лице в веб-интерфейсе

После настройки Liveness-детектора для каждого события будет отображаться оценка liveness обнаруженного лица.



Примечание: Liveness может принимать значение `null`, если детектор живых лиц отключен или на предоставленном изображении невозможно достоверно оценить Liveness.

Используйте фильтр *Liveness* для просмотра событий только с живыми людьми или только с изображениями, если имели место спуфинговые атаки.



См.также:

Liveness как автономный сервис

2.7.5 Liveness как автономный сервис

См.также:

Распознавание живых лиц в реальном времени (Liveness)

Помимо *интегрированной* антиспуфинговой системы, отличающей живые лица от их изображений, FindFace Multi предоставляет сервис определения живых лиц на основе API `findface-liveness-api`.

Сервис `findface-liveness-api` берет определенное количество кадров из предоставленного видеофрагмента и возвращает изображение лица наилучшего качества вместе с десятичным результатом `liveness` по этому лицу, усредненным по взятым кадрам. Если заданы соответствующие настройки, сервис также может возвращать полнокадровые и нормализованные изображения лиц и сохранять результат детекции в кэше `findface-sf-api`, возвращая `detection_id`.

Сервис `findface-liveness-api` автоматически устанавливается и активируется, так как FindFace Multi использует его для аутентификации *по лицу*.

Вы можете установить и использовать сервис `findface-liveness-api` отдельно от FindFace Multi. Об этом и идет речь в настоящем разделе.

В этом разделе:

- Установка и настройка `findface-liveness-api`
- Запросы HTTP API к сервису `findface-liveness-api`

Установка и настройка `findface-liveness-api`

Для того чтобы установить сервис отдельно от FindFace Multi, установите *APT-репозиторий* FindFace Multi и выполните следующие команды:

```
sudo apt update
sudo apt install findface-liveness-api
```

Вы можете настроить параметры `findface-liveness-api` в файле конфигурации `/etc/findface-liveness-api.ini`:

```
sudo vi /etc/findface-liveness-api.ini
```

```
listen: :18301
```

(continues on next page)

(продолжение с предыдущей страницы)

```

liveness-threshold: 0.95
fullframe-jpeg-quality: 75
max-decoded-frames: 30
min-selected-frames: 10
mf-selector: reject
extraction-api:
  request-batch-size: 16
  timeouts:
    connect: 5s
    response_header: 30s
    overall: 35s
    idle_connection: 10s
  extraction-api: http://127.0.0.1:18666
sf-api:
  timeouts:
    connect: 5s
    response_header: 30s
    overall: 35s
    idle_connection: 10s
  sf-api: http://127.0.0.1:18411
limits:
  video-size: 10485760
  video-length-sec: 60
  video-fps: 30
  video-width-px: 1920
  video-height-px: 1080

```

Параметр	Описание
fullframe-jpeg-quality	Количество JPEG полных кадров в поле photo.
max-decoded-frames	Завершить декодирование после получения указанного количества кадров.
min-selected-frames	Минимальное количество финальных кадров, успешно прошедших декодирование и извлечение liveness. Должно быть равным или меньшим max-decoded-frames.
mf-selector	Поведение сервиса при наличии нескольких лиц на кадре видео: reject - отклонить данный кадр, biggest - использовать самое крупное лицо для определения liveness.
extraction-api -> request-batch-size	Размер пакета с кадрами для извлечения liveness.
limits -> video-size	Максимальный размер видео, байты.
limits -> video-length-sec	Максимальная длина видео, секунды.
limits -> video-fps	Максимальное количество FPS видео.
limits -> video-width-px	Максимальная ширина видео, пиксели.
limits -> video-height-px	Максимальная высота видео, пиксели.

Для того чтобы запустить сервис findface-liveness-api и добавить его в автозагрузку, выполните

КОМАНДЫ:

```
sudo systemctl start findface-liveness-api.service && sudo systemctl enable findface-
↪liveness-api.service
```

Запросы HTTP API к сервису findface-liveness-api

Для взаимодействия с сервисом `findface-liveness-api` используйте запросы HTTP API. В примере ниже запрос POST отправляется со следующими опциональными параметрами:

- `return_detection` (default=False): сохранить лучшее лицо в кэше `findface-sf-api` и вернуть его `detection_id`.
- `return_normalized` (default=False): вернуть нормализованное изображение лица в поле `normalized`.
- `return_photo` (default=False): вернуть полный кадр в поле `photo`.

Пример

Запрос

```
curl -i -X POST \
  'http://127.0.0.1:18301/v1/video-liveness?return_detection=true&return_normalized=true&
↪return_photo=true' \
  -H 'Content-Type: video/mp4' \
  --data-binary '@/home/my_video.mp4'
```

Ответ

```
HTTP/1.1 100 Continue
HTTP/1.1 200 OK
Content-Type: application/json
X-Request-Id: LA:WSP2NcHc
Date: Mon, 07 Sep 2020 15:30:05 GMT
Transfer-Encoding: chunked
{
  "alive": true,
  "average_liveness": 0.9706386,
  "best_face": {
    "liveness": 0.97768883,
    "quality": 0.89638597,
    "bbox": {
      "left": 0,
      "top": 578,
      "right": 307,
      "bottom": 1154
    },
    "detection_id": "btb53vbp688s1njt3bv0",
    "photo": "/9j/2wCEAAgGBgcGBQgHBwcJ...",
    "normalized": "iVBORwOKGgoAAAANSUHEU...",
```

(continues on next page)

(продолжение с предыдущей страницы)

```
        "frame_no": 1,  
        "frame_ts": 0.033275817  
    }  
}
```

См.также:

Деактивация сервиса findface-liveness-api, установленного вместе с FindFace Multi

2.7.6 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 *Метки* 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 экземпляра `/etc/findface-video-worker-cpu.ini` (`/etc/findface-video-worker-gpu.ini`) и укажите в нем заданные метки в формате `имя_метки=true` (`terminal_1` в примере ниже).

```
sudo vi /etc/findface-video-worker-cpu.ini  
sudo vi /etc/findface-video-worker-gpu.ini  
  
labels = terminal_1=true
```

5. Перезапустите `findface-video-worker`.

```
sudo systemctl restart findface-video-worker-cpu.service  
sudo systemctl restart findface-video-worker-gpu.service
```

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

Предупреждение: Если камера с меткой обрабатывается экземпляром `findface-video-worker(-gpu)` без меток и появляется свободный экземпляр с меткой, камера автоматически на него не переключится. Чтобы переключить камеру, перезапустите экземпляр `findface-video-worker(-gpu)` с меткой.

2.7.7 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 автомобилей выполните следующие действия:

1. Подготовьте список пользовательских вкладок и полей для добавления в карточки.
2. Откройте файл конфигурации `/etc/findface-security/config.py`.

```
sudo vi /etc/findface-security/config.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. Если не задана, поле появится на главной странице карточки (той, что с фотографией).
 - `'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 вкладок в карточке. Первая вкладка в списке соответствует главной странице карточки.

```
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
      },
      {
        '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
      }
    ]
  }
}
```

(continues on next page)

(продолжение с предыдущей страницы)

```

    },
    {
      '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 автомобилей. Для этого продублируйте содержимое секции `human_card` в секцию `car_card` и измените его по аналогии.
5. Перезапустите сервис `findface-security`.

```
sudo systemctl restart findface-security.service
```

Вы увидите, что в карточках появилось новое содержимое.

2.7.8 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 автомобилей

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 `/etc/findface-security/config.py`.

```
sudo vi /etc/findface-security/config.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;
- `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
                },
                {
                    "field_name": "tag_name_3",
                    "type": "bool",
                    "default": True
                }
            ]
        }
    }
```

(continues on next page)

(продолжение с предыдущей страницы)

```

    },
  },
}

```

4. Добавьте новые метаполя в структуру базы данных векторов признаков.
5. Перезапустите сервис `findface-security`.

```
sudo systemctl restart findface-security.service
```

Вы можете работать с новыми метаполями через *HTTP API*, используя методы `objects/faces/`.

Настройка метаполей для силуэтов и автомобилей

Назначение пользовательских метаполей силуэтам и автомобилям 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",
        "default": True
      }
    ]
  }
  'car_object': {
    'items': [
      {
        "field_name": "tag_name_1",

```

(continues on next page)

(продолжение с предыдущей страницы)

```

        "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`.

См.также:

Для создания пользовательских вкладов, полей и фильтров в карточках см. *Пользовательские вкладки, поля и фильтры в карточке*.

2.7.9 Пакетная загрузка фотографий через консоль

Помимо *веб-интерфейса* для пакетной загрузки фотографий в картотеку можно использовать консольную утилиту `findface-security-uploader`. Рекомендуется предпочесть данную утилиту веб-интерфейсу, если количество загружаемых фотографий превышает 10 000.

Предупреждение: В текущей версии утилита `findface-security-uploader` не поддерживает автомобили и силуэты, только лица.

Совет: Для вызова справки `findface-security-uploader` выполните команду:

```

findface-security-uploader --help

Usage: findface-security-uploader [OPTIONS] COMMAND [ARGS]...

Options:
  --job PATH           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

```

(continues on next page)

(продолжение с предыдущей страницы)

```
print Print contents of job file as JSON
run    Run upload job
```

```
findface-security-uploader add --help
```

```
Usage: findface-security-uploader 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.
```

```
findface-security-uploader print --help
```

```
Usage: findface-security-uploader 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.
```

```
findface-security-uploader 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]
--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
--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 или команду `find`.

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 указанного каталога (`/home/user/25_celeb/` в примере) выполните следующую команду:

```
python3 tsv_builder.py /home/user/25_celeb/
```

Пример использования команды `find`:

```
find photos/ -type f -iname '*g' | while read x; do y="${x%.*}"; printf "%s\t%s\n" "  
↪$x" "${y##*/}"; done
```

- Создайте файл задания (job-файл) из CSV- или TSV-файла, используя метод `add` утилиты. В результате в текущем каталоге будет создан файл `enroll-job.db`.

```
findface-security-uploader add images.tsv
```

Опции `add`:

- `--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`.

- Выполните задание `job`, используя метод `run` утилиты.

```
findface-security-uploader run --watch-lists 2 --api http://127.0.0.1:80 --user␣  
↪admin --password password
```

Важные опции `run`:

- `--parallel`: количество потоков загрузки фотографий, по умолчанию 10. Чем больше потоков, тем быстрее будет завершена загрузка, однако также потребуется и большее количество ресурсов.
- `--all-faces`: загрузить все лица, если на фотографии их несколько.
- `--api`: URL API 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.
- `--failed`: в случае неудачи при обработке job-файла исправьте ошибку и повторите попытку с данной опцией.

- `--inactive`: отметить новые карточки как неактивные.
 - `--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.
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.

```
findface-security-uploader print --failed
findface-security-uploader print --noface
```

2.7.10 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 *Дедуплицировать события* 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.

2.7.11 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 `/etc/findface-security/config.py`.

```
sudo vi /etc/findface-security/config.py
```

2. Отключите сохранение событий без совпадений, установив `'IGNORE_UNMATCHED': True`.

```
...

FFSECURITY = {
    ...

    # do not save unmatched events (GDPR support)
    'IGNORE_UNMATCHED': False,

    ...
}
```

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': False,

    # 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 -> available_video_wall_features.

```
FFSECURITY_UI_CONFIG = {
    ...
    "available_video_wall_features": {
        ...
        "gdpr": True
    }
}
```

5. Перезапустите findface-security.

```
sudo systemctl restart findface-security.service
```

См.также:

Видеостена

2.7.12 Настройка Видеомегнитофона

Внутренняя *архитектура* Ядра FindFace позволяет встроить Видеомегнитофон — дополнительный функционал, который записывает, хранит и проигрывает видеоданные с камер.

Данный раздел посвящен настройке функционала Видеомегнитофон.

Примечание: Если вы еще не развернули данную систему, сделайте это с помощью *консольного инсталлятора* FindFace Multi или *пошаговых инструкций*.

В этом разделе:

- Включение Видеомегнитофона
- Особенности отключения Видеомегнитофона
- Удаление видео вручную

Включение Видеомегнитофона

Для активации Видеомегнитофона выполните следующие действия:

1. Включите возможность передачи видеофрагментов от findface-video-worker в сервис findface-video-storage.

Важно: Данная настройка делает функционал Видеомегнитофона доступным через *HTTP API*. Она также обязательна, если вы собираетесь использовать Видеомегнитофон в составе *веб-интерфейса* FindFace Multi.

Выполните следующие действия:

1. Откройте файл конфигурации `/etc/findface-video-worker-cpu.ini` (`/etc/findface-video-worker-gpu.ini`) для каждого экземпляра `findface-video-worker`, который будет поставлять видео в Видеомэгнитофон.

```
sudo vi /etc/findface-video-worker-cpu.ini
sudo vi /etc/findface-video-worker-gpu.ini
```

2. Установите `enabled = true` в разделе `recorder`.

```
#-----
[recorder]
#-----
## video recording enabled
## type:bool env:CFG_RECORDER_ENABLED longopt:--recorder-enabled
enabled = true
```

3. Перезапустите каждый задействованный экземпляр `findface-video-worker`.

```
sudo systemctl restart findface-video-worker*
```

2. Настройте Видеомэгнитофон для работы в составе веб-интерфейса FindFace Multi.

Примечание: Пропустите следующие шаги, если вам не нужно, чтобы инструменты Видеомэгнитофона отображались в веб-интерфейсе FindFace Multi.

Выполните следующие действия:

1. Откройте файл конфигурации `/etc/findface-security/config.py`.

```
sudo vi /etc/findface-security/config.py
```

2. Найдите раздел `FFSECURITY_UI_CONFIG` -> `vms`.

Список параметров для настройки:

Параметр	Описание
"enabled"	Установите True, чтобы добавить флажок <i>Включить запись в настройки камер</i> .
"video_player"	Установите True, чтобы включить отображение видеоплеера в <i>уведомлениях о событиях</i> и <i>предварительном просмотре камеры</i> (вместо статических кадров). Данная настройка требует установленного "enabled": True.
"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" -> "faces" -> "enabled"	Установите True, чтобы отмечать на временной шкале все события с лицами в виде маркеров.
"timeline" -> "objects" -> "faces" -> "limit"	Максимальное количество событий с лицами, одновременно отмеченных на временной шкале. Если событий с лицами больше, вам будет предложено увеличить масштаб.
"timeline" -> "objects" -> "cars" -> "enabled"	Установите True, чтобы отмечать на временной шкале все события с автомобилями в виде маркеров.
"timeline" -> "objects" -> "cars" -> "limit"	Максимальное количество событий с автомобилями, одновременно отмеченных на временной шкале. Если событий с автомобилями больше, вам будет предложено увеличить масштаб.

```
FFSECURITY_UI_CONFIG = {
    "vms": {
        "enabled": True,
        "video_player": True,
        "timeline": {
            "min_zoom": 0.2,
            "max_zoom": 200,
            "objects": {
                "events": {
                    "faces": {
                        "enabled": True,
                        "limit": 500,
                    },
                    "cars": {
                        "enabled": False,
                        "limit": 500,
                    },
                },
            },
        },
    },
}
```

3. Установите максимальное время хранения записанных видеофрагментов, используя параметр VMS_VIDEOS_MAX_AGE. Видеофрагменты старше указанного количества дней будут ав-

томатически удаляться.

Совет: Чтобы приблизительно оценить требуемое время хранения, учтите, что камера 1920×1080 производит около 60 ГБ видео в день. Мы рекомендуем измерить точный объем видео, поступающий с ваших камер, и рассчитать необходимое время хранения на основании данного значения и доступного места на диске.

Совет: Если вы собираетесь хранить видеофрагменты в течение длительного времени, обязательно выполняйте мониторинг оставшегося свободного пространства, используя Zabbix, Prometheus или аналогичную программу. Это не позволит видео занять все свободное место, тем самым заблокировав работу сервера.

```
# VMS video chunks will be deleted older than VMS_VIDEOS_MAX_AGE
'VMS_VIDEOS_MAX_AGE': 7,
```

4. Перезапустите сервис `findface-security`.

```
sudo systemctl restart findface-security.service
```

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 `/etc/findface-security/config.py` и `/etc/findface-video-worker-*.ini`.

Удаление видео вручную

Вы можете вручную удалить видеофрагменты старше заданного количества дней, выполнив приведенную ниже команду. Для задания количества дней используйте аргумент `--vms-videos-max-age`.

```
sudo findface-security cleanup_vms --vms-videos-max-age=30
```

См.также:

- *Пошаговое развертывание функционала Видеомагнитофон*
- *Включение видеозаписи с камеры*
- *Видеоплеер в уведомлении о событии*
- *Резервное копирование и восстановление FindFace Multi и всех данных*
- *Резервное копирование и восстановление базы данных Видеомагнитофона (MongoDB)*

2.7.13 Аутентификация по сертификатам КриптоПро

По умолчанию вход пользователей в систему осуществляется по логину и паролю. При необходимости можно включить аутентификацию по сертификатам КриптоПро.

В этом разделе:

- *Настройка аутентификации по сертификатам КриптоПро*
- *Загрузка сертификата пользователя*

Настройка аутентификации по сертификатам КриптоПро

Для настройки аутентификации по сертификатам КриптоПро выполните следующие действия:

1. Загрузите на сервер FindFace Multi архивы КриптоПро CSP 4.0 для Linux (x64, deb) и КриптоПро ЭЦП SDK 2.0 Linux x64.
2. Распакуйте загруженные архивы.
3. Из директории `/opt/findface-security/lib/python3.6/site-packages/ffsecurity_cproauth` запустите скрипт `build.sh`.

```
chmod +x build.sh
sudo ./build.sh
```

По требованию скрипта установите дополнительные deb-пакеты из распакованных архивов.

```
Please install lsb-cprocsp-devel from CryptoPro CSP ( https://www.cryptopro.ru/products/csp/downloads )

apt install lsb-cprocsp-devel
...
```

4. Откройте файл конфигурации `findface-security`.

```
sudo vi /etc/findface-security/config.py
```

5. Добавьте/раскомментируйте следующие настройки:

```
...
UVICORN_SETTINGS = {
...
'proxy_headers': True
}
...
# ===== CryptoPRO authentication =====
INSTALLED_APPS.append('ffsecurity_cproauth')
REST_FRAMEWORK['DEFAULT_AUTHENTICATION_CLASSES'] = [
    'ffsecurity.auth.TokenAuthentication',
    'ffsecurity_cproauth.auth.CryptoProOrTokenAuthentication'
]
```

6. Перезапустите `findface-security`.

```
sudo systemctl restart findface-security.service
```

На этом настройка аутентификации по сертификатам КриптоПро будет завершена. Можно приступить к загрузке сертификатов в систему.

Загрузка сертификата пользователя

Для того чтобы загрузить сертификат пользователя в систему, выполните следующие действия:

1. Перейдите на вкладку *Настройки*. Выберите *Пользователи*.
2. Откройте карточку пользователя.
3. Нажмите *Добавить сертификат*.

2.8 Обслуживание и устранение неисправностей

2.8.1 Обновление до FindFace Multi 1.2

Совет: Если вы используете наш продукт FindFace Security, развернутый в Ubuntu 18.04, сначала **обновите** его до FindFace Multi 1.0, а затем до FindFace Multi 1.2.

Для обновления FindFace Multi с ранних версий до версии 1.2 выполните следующие действия:

1. Откройте файл конфигурации `/etc/findface-security/config.py` (`/etc/ffsecurity/config.py`). Сохраните для последующего использования значения следующих параметров: `EXTERNAL_ADDRESS`, `SECRET_KEY`, `VIDEO_DETECTOR_TOKEN`, `ROUTER_URL`.

```
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',
    ...
}
```

2. В FindFace Multi 1.2 мы значительно усовершенствовали внутреннюю структуру сущности **report** (отчет). Однако у модернизированной сущности отсутствует обратная совместимость с более ранними версиями продукта. Перед обновлением обязательно загрузите все нужные вам отчеты с вкладки *Отчеты*. Позже, в обновленной системе, вы сможете использовать их в качестве образцов для воссоздания ранее существовавших отчетов.
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`.

Совет: Подробнее см. *Резервное копирование и восстановление FindFace Multi и всех данных*.

```
sudo mkdir -p /etc/findface_dump
cd /etc/findface_dump
sudo findface-storage-api-dump -config /etc/findface-sf-api.ini
```

5. Установите apt-репозиторий с версией FindFace Multi, используя консольный инсталлятор согласно инструкции в *этом разделе*.

Совет: На этапе установки моделей нейронных сетей (APT repository doesn't include recognition models. Do you want to install them now?) используйте один из следующих вариантов:

- Выберите No
- Выберите Yes и оставьте выбранными все модели нейронных сетей. Введите Done

Чтобы решить, какой вариант подходит вам лучше всего, ознакомьтесь с комментариями к шагу № 9.

6. Установите NATS, добавьте этот сервис в автозагрузку и запустите.

Важно: Пропустите этот шаг при обновлении с FindFace Multi 1.1.

```
sudo apt install -y nats-server
sudo systemctl enable nats-server
sudo systemctl restart nats-server
```

7. Установите из репозитория сервисы FindFace Multi в соответствии со своей архитектурной схемой.
CPU-версия:

```
sudo apt update
sudo apt install findface-security findface-security-ui findface-extraction-api
↪ findface-ntls findface-sf-api findface-tarantool-server findface-upload findface-
↪ video-manager findface-video-worker-cpu findface-counter findface-liveness-api
```

GPU-версия:

```
sudo apt update
sudo apt install findface-security findface-security-ui findface-extraction-api-gpu
↪ findface-ntls findface-sf-api findface-tarantool-server findface-upload findface-
↪ video-manager findface-video-worker-gpu findface-counter findface-liveness-api
```

Важно: Для работы FindFace Multi на GPU требуется предварительная установка *драйверов*

NVIDIA.

Важно: В какой-то момент вам будет предложено выбрать, какую версию конфигурационного файла `findface-security` использовать. Выберите вариант `Install the packages maintainer's version` (использовать версию из пакета). Сделайте то же самое, если аналогичное сообщение появилось для другого компонента (`findface-extraction-api`, `findface-video-worker` и т. д.). Во всех случаях имеет смысл предпочесть более новую версию файла конфигурации.

- Откройте файл конфигурации `/etc/findface-security/config.py` и вставьте в него сохраненные значения параметров `EXTERNAL_ADDRESS`, `SECRET_KEY`, `VIDEO_DETECTOR_TOKEN` и `ROUTER_URL`. Заполните раздел `DATABASES` по аналогии: `'PORT': 5439`, `'USER': 'ntech'`, `'PASSWORD': '9T3g1nXy9yx3y8MIGm9fbef3dia8UTc'` (пароль из `/etc/pgbouncer/userlist.txt`).

```
sudo vi /etc/findface-security/config.py
...
# Database is used by FindFace Security to store cameras,
# camera groups, watchlists and so on. Only PostgreSQL is supported.
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'DISABLE_SERVER_SIDE_CURSORS': True,
        'NAME': 'ffsecurity',
        'PORT': 5439, 'USER': 'ntech', 'PASSWORD': '9T3g1nXy9yx3y8MIGm9fbef3dia8UTc3
↵',
    }
}
...
# 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',
```

- Откройте старую версию файла конфигурации `findface-ntls`, доступную по адресу `/etc/findface-ntls.cfg.dpkg-old`, и сравните ее с новой версией `/etc/findface-ntls.cfg`. 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.ucf-old` на соответствие `/etc/findface-extract-api.ini` и т. д.

```
sudo vi /etc/findface-ntls.cfg.dpkg-old
sudo vi /etc/findface-ntls.cfg
```

(continues on next page)

(продолжение с предыдущей страницы)

```
sudo vi /etc/findface-extraction-api.ini.ucf-old
sudo vi /etc/findface-extraction-api.ini

...
```

Важно: Имейте в виду, что новая версия файла конфигурации `findface-extraction-api` по умолчанию содержит модель `liveness.colombo`, которая отсутствовала в прежних версиях продукта. Это может потенциально привести к сбою системы. Существуют следующие способы предотвратить сбой:

- Восстановите для параметра `face_liveness` предыдущее значение: `liveness.alleyn.v2.cpu.fnk/liveness.alleyn.v2.gpu.fnk`
- Установите модель `liveness.colombo.cpu.fnk/liveness.colombo.gpu.fnk` на шаге №5

10. Откройте новую версию файла конфигурации `/etc/findface-video-worker-cpu.ini` (`/etc/findface-video-worker-gpu.ini`). В разделе `streamer` установите `tracks = true` и `tracks_last = true`, чтобы рамки с объектами на видеостене гарантированно отображались правильно.

```
sudo vi /etc/findface-video-worker-cpu.ini
sudo vi /etc/findface-video-worker-gpu.ini

[streamer]
#-----
...

## use tracks instead detects for streamer
## type:bool env:CFG_STREAMER_TRACKS longopt:--streamer-tracks
tracks = true

## use tracks with lastFrameId=currentFrameId (.tracks must be true)
## type:bool env:CFG_STREAMER_TRACKS_LAST longopt:--streamer-tracks-last
tracks_last = true
```

11. Перезагрузите сервисы.

```
sudo systemctl restart findface-ntls findface-extraction-api findface-video-worker*
↪ findface-video-manager findface-sf-api findface-counter findface-liveness-api
```

12. Измените структуру базы данных Tarantool с помощью файла `tnt_schema.lua` из FindFace Multi.

```
sudo findface-security make_tnt_schema | sudo tee /etc/findface-security/tnt_schema.
↪ lua
```

13. Остановите шарды `findface-tarantool-server`. Удалите данные из каталогов, соответствующих активным шардам.

```
sudo systemctl stop 'tarantool@*'

sudo rm /opt/ntech/var/lib/tarantool/shard-*/{index,snapshots,xlogs}/*
```

14. Перейдите в директорию с файлами конфигурации Tarantool `/etc/tarantool/instances.available/`. Проверьте, содержит ли каждый файл конфигурации `shard-*.lua` команду `dofile`

и определения `meta_indexes` и `meta_scheme`, как показано в примере ниже.

```
sudo vi /etc/tarantool/instances.available/shard-*.lua

...
dofile("/etc/findface-security/tnt_schema.lua")
...
FindFace.start("127.0.0.1", 8101, {
    license_ntls_server="127.0.0.1:3133",
    meta_indexes=meta_indexes,
    meta_scheme = meta_scheme
})
```

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
```

16. Восстановите базу данных Tarantool из резервной копии.

```
cd /etc/findface_dump

for x in *.json; do sudo findface-storage-api-restore -config /etc/findface-sf-api.
↪ini < "$x"; done
```

17. Перенесите схему базы данных из FindFace Multi в PostgreSQL, заново создайте *предустановленные* пользовательские роли и первого администратора.

```
sudo findface-security migrate
sudo findface-security create_groups
sudo findface-security create_default_user
```

18. Перезапустите сервисы `findface-security` и `nginx`.

```
sudo systemctl restart findface-security.service nginx.service
```

19. Воссоздайте ранее существовавшие отчеты.

Важно: Следующие вкладки были усовершенствованы и переименованы в FindFace Multi 1.2:

- *Досье* превратились в *Карточки*
 - *Эпизоды* переименованы в *Эпизоды -> Люди*
 - *Персоны* превратились в *Кластеры*
-

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
```

(continues on next page)

(продолжение с предыдущей страницы)

```
sudo systemctl disable apt-daily.timer
sudo systemctl disable apt-daily.service
sudo systemctl daemon-reload
```

Примечание: Вы можете дополнительно развернуть Видеомэгнитофон. См. *Пошаговое развертывание функционала Видеомэгнитофона*.

2.8.2 Резервное копирование и восстановление FindFace Multi и всех данных

Этот раздел посвящен резервному копированию и восстановлению хранилищ данных FindFace Multi и вашей системы в целом.

В этом разделе:

- *Список хранилищ*
- *Резервное копирование и восстановление базы данных векторов признаков (Tarantool)*
 - *Утилиты*
 - *Резервное копирование базы данных*
 - *Восстановление базы данных*
- *Резервное копирование и восстановление основной базы данных (PostgreSQL)*
- *Резервное копирование и восстановление артефактов (findface-upload)*
- *Резервное копирование и восстановление базы данных Видеомэгнитофона (MongoDB)*
- *Резервное копирование и восстановление настроек*
- *Резервное копирование и восстановление системы целиком*

Список хранилищ

FindFace Multi использует следующие хранилища данных:

- База данных векторов признаков на основе Tarantool, в которой хранятся векторы признаков и события.
- Основная база данных системы на основе PostgreSQL, в которой хранятся внутренние данные системы, карточки, учетные записи пользователей и настройки камер.
- Каталог `/var/lib/findface-security/uploads`, в котором хранятся загруженные в карточки фотографии, видеофайлы, полные кадры событий и счетчиков, а также миниатюры объектов.
- Каталог `/var/lib/ffupload/`, в котором хранятся такие артефакты событий, как нормализованные изображения объектов.

Примечание: В данном каталоге также хранятся видеофрагменты, если в системе активирована функция *Видеомагнитофон*.

- (Только вместе с Видеомагнитофоном) База данных на основе MongoDB, в которой хранится метаданная о видеофрагментах, включая их точное местоположение в хранилище `/var/lib/ffupload/`.

Резервное копирование и восстановление базы данных векторов признаков (Tarantool)

В базе данных векторов признаков на основе Tarantool 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 автомобилей.
- `ffsec_car_objects`: векторы признаков, извлеченные из загруженных в картотеку изображений автомобилей.
- `ffsec_car_clusters`: центроиды кластеров автомобилей.
- `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.

Функционал резервного копирования и восстановления базы данных позволяет при необходимости восстанавливать содержимое данных галерей.

Во избежание потери данных рекомендуется создавать резервную копию базы данных векторов признаков не реже одного раза в неделю. В целом частота резервного копирования зависит от количества карточек и событий распознавания объектов, а также доступного дискового пространства.

Резервную копию базы данных также нужно обязательно создать перед *миграцией* системы на другую модель нейронной сети.

Утилиты

Для резервного копирования и восстановления базы данных векторов признаков FindFace Multi необходимы следующие утилиты:

1. резервное копирование: `findface-storage-api-dump`,
2. восстановление: `findface-storage-api-restore`.

Данные утилиты автоматически устанавливаются вместе с компонентом `findface-sf-api`.

Резервное копирование базы данных

Для резервного копирования базы данных векторов признаков используйте утилиту `findface-storage-api-dump` следующим образом:

Важно: Сервисы `findface-tarantool-server` и `findface-sf-api` должны быть активны.

Примечание: Резервное копирование можно также применить к распределенной базе данных. В этом случае утилита `findface-storage-api-dump` создаст резервные копии всех шардов, указанных в `/etc/findface-sf-api.ini`.

1. На сервере с установленным `findface-sf-api` создайте каталог для хранения резервных копий (`/etc/findface_dump` в примере ниже).
2. Запустите утилиту `findface-storage-api-dump` следующей командой:

```
sudo findface-storage-api-dump -output-dir=/etc/findface_dump -config /etc/findface-
→sf-api.ini
```

Утилита создаст резервные копии всех галерей и запишет их в указанный каталог в виде файлов с соответствующими именами (`ffsec_body_events.json`, `ffsec_face_events` и т. д.). Эти файлы содержат все данные, необходимые для полного восстановления галерей.

Восстановление базы данных

Для восстановления базы данных векторов признаков запустите утилиту `findface-storage-api-restore` для всех файлов в директории с резервными копиями:

```
sudo findface-storage-api-restore -config /etc/findface-sf-api.ini /etc/findface_dump/*.
→json
```

Процесс восстановления можно при необходимости прервать с сохранением выполненной работы. Для того чтобы продолжить процесс после прерывания, снова запустите утилиту `findface-storage-api-restore`.

См. также:

- [Опции резервного копирования базы данных](#)
- [Опции восстановления базы данных](#)

Резервное копирование и восстановление основной базы данных (PostgreSQL)

Для резервного копирования основной базы данных `ffsecurity` на PostgreSQL выполните команду:

```
sudo -u postgres pg_dump ffsecurity > ffsecurity_postgres_backup.sql
```

Для того чтобы восстановить основную базу данных, выполните следующие действия:

1. Остановите сервис `findface-security`.

```
sudo systemctl stop findface-security.service
```

2. Остановите сервис `pgbouncer`, чтобы удалить активные сессии с базой данных `ffsecurity`.

```
sudo systemctl stop pgbouncer.service
```

3. Откройте интерактивный терминал PostgreSQL.

```
sudo -u postgres psql
```

4. Удалите прежнюю базу данных `ffsecurity`.

```
DROP DATABASE ffsecurity;
```

5. Создайте новую базу `ffsecurity`. Выйдите из интерактивного терминала PostgreSQL.

```
CREATE DATABASE ffsecurity WITH OWNER ntech ENCODING 'UTF-8' LC_COLLATE='C.UTF-8'
↳LC_CTYPE='C.UTF-8' TEMPLATE template0;
```

6. Запустите сервис `pgbouncer`.

```
sudo systemctl start pgbouncer.service
```

7. Восстановите содержимое базы данных из резервной копии.

```
sudo -u postgres psql -d ffsecurity -f ffsecurity_postgres_backup.sql
```

8. Перенесите схему базы данных из FindFace Multi в PostgreSQL, заново создайте группы пользователей с *предустановленными* правами и первого пользователя с правами администратора.

```
sudo findface-security migrate
sudo findface-security create_groups
sudo findface-security create_default_user
```

9. Запустите сервис `findface-security`.

```
sudo systemctl start findface-security.service
```

Резервное копирование и восстановление артефактов (`findface-upload`)

Артефакты FindFace Multi, такие как загруженные фотографии, видеофайлы и артефакты событий (полные кадры, миниатюры объектов и нормализованные изображения объектов) хранятся в следующих каталогах:

- `/var/lib/findface-security/uploads`
- `/var/lib/ffupload/`

Примечание: Директория `/var/lib/ffupload/` также хранит видеофрагменты, если в вашей системе есть работающий экземпляр Видеомагнитофона.

Примечание: Обе директории управляются компонентом `findface-upload`.

Для резервного копирования артефактов выполните команды:

```
sudo tar -cvzf /home/some_directory/var_lib_ffsecurity_uploads.tar.gz /var/lib/findface-
security/uploads/
sudo tar -cvzf /home/some_directory/var_lib_ffupload.tar.gz /var/lib/ffupload/
```

Для того чтобы восстановить артефакты, перейдите в корневую директорию и выполните команды:

```
cd /
sudo tar -xvf /home/some_directory/var_lib_ffsecurity_uploads.tar.gz
sudo tar -xvf /home/some_directory/var_lib_ffupload.tar.gz
```

Резервное копирование и восстановление базы данных Видеомagniтофона (MongoDB)

MongoDB — это стороннее программное обеспечение, реализующее базу данных Видеомagniтофона. Эта база данных хранит метаданные о видеофрагментах, включая их точное местонахождение в хранилище `/var/lib/ffupload/`.

Для резервного копирования MongoDB выполните команду:

```
sudo mongodump -o=/home/some_directory/mongodb_backup
```

Для восстановления MongoDB выполните команду:

```
sudo mongorestore /home/some_directory/mongodb_backup
```

Резервное копирование и восстановление настроек

Полный набор файлов конфигурации FindFace Multi, включая структурную схему Tarantool, автоматически резервируется при *удалении экземпляра продукта*. Резервная копия хранится в директории `~/ffmulti_bak_${datetime}/etc/`.

При переустановке FindFace Multi настройки должны быть восстановлены после завершения *консольной установки* или сразу после установки сервисов из *APT-репозитория*.

Полный набор резервных копий файлов следующий:

```
ls -R -p
.:
findface-counter.ini.bak      findface-liveness-api.ini.bak  findface-security/      □
findface-video-manager.conf.bak findface-video-streamer-cpu.ini.bak nginx/
findface-extraction-api.ini.bak findface-ntls.cfg.bak         findface-sf-api.ini.bak □
findface-video-storage.conf.bak findface-video-worker-cpu(gpu).ini.bak

./findface-security:
config.py  tnt_schema.lua

./nginx:
sites-enabled/

./nginx/sites-enabled:
ffsecurity-nginx.conf
```

Для того чтобы восстановить настройки FindFace Multi, выполните следующие действия:

1. Откройте файл конфигурации `/etc/findface-security/config.py` нового экземпляра FindFace Multi. Найдите параметр `DATABASES -> default -> PASSWORD`, в котором хранится пароль пользователя `ntech` от базы данных `findface-security PostgreSQL`. Скопируйте и вставьте его в резервную копию `~/ffmulti_bak_${datetime}/etc/findface-security/config.py`.

```
sudo vi /etc/findface-security/config.py

DATABASES = {
    'default':
        'PASSWORD': 'some_pass'
```

2. В директории `~/ffmulti_bak_${datetime}/etc/` удалите из файлов расширение `.bak` любым доступным способом. Например, вы можете выполнить следующую команду.

```
sudo rename 's/.ini.bak/.ini/' * && sudo rename 's/.conf.bak/.conf/' * && sudo
↪rename 's/.cfg.bak/.cfg/' *
```

Совет: Если в вашей системе отсутствует команда `rename`, можно установить ее следующим образом:

```
sudo apt install rename
```

3. Рекурсивно скопируйте файлы резервных копий в каталог `/etc`.

```
sudo cp -r * /etc
```

4. Измените структуру базы данных с помощью исходного файла `tnt_schema.lua`.

```
sudo findface-security make_tnt_schema | sudo tee /etc/findface-security/tnt_schema.
↪lua
```

5. Перезапустите сервисы.

На CPU:

```
sudo systemctl restart findface-counter findface-liveness-api findface-video-
↪manager findface-extraction-api findface-ntls findface-sf-api findface-video-
↪worker-cpu findface-security
```

На GPU:

```
sudo systemctl restart findface-counter findface-liveness-api findface-video-
↪manager findface-extraction-api findface-ntls findface-sf-api findface-video-
↪worker-gpu findface-security
```


Сервисы Видеомониторинга:

```
sudo systemctl restart findface-video-streamer-cpu findface-video-storage
```

Резервное копирование и восстановление системы целиком

Если вы намерены создать резервную копию FindFace Multi перед его удалением, будет достаточно следовать пошаговым инструкциям в разделе *Удаление экземпляра продукта*. Предоставленный скрипт `ffmulti_uninstall.sh` может автоматически создать в директории `~/ffmulti_bak_${datetime}/` резервные копии файлов конфигурации FindFace Multi и всех хранилищ данных.

Для восстановления FindFace Multi после его удаления, используйте следующий алгоритм:

1. *Разверните FindFace Multi.*
2. *Восстановите настройки из резервных копий файлов конфигурации.*
3. *Восстановите Tarantool.*
4. *Восстановите PostgreSQL.*
5. *Восстановите артефакты системы.*
6. *Восстановите MongoDB (если вы используете Видеомониторинг).*

2.8.3 Добавление пользовательского языка

По умолчанию в веб-интерфейсе для выбора доступны только английский и русский языки. Другие языки предоставляются по запросу (support@ntechlab.com) и добавляются в систему через файл конфигурации `/etc/findface-security/config.py`.

Для добавления пользовательского языка в систему выполните следующие действия:

1. Загрузите файл с локализацией, предоставленный нашими специалистами, в директорию `/usr/share/ffsecurity-ui/ui-static/` на центральном сервере FindFace Multi.
2. Откройте файл конфигурации `/etc/findface-security/config.py`.

```
sudo vi /etc/findface-security/config.py
```

3. В раздел `FFSECURITY_UI_CONFIG` вставьте секцию `languages` и заполните ее по аналогии с примером ниже.

```
FFSECURITY_UI_CONFIG = {
    ...
    "languages": {
        "items": [
            {
                "name": "es",
                "label": "Español",
                "url": "/ui-static/es_i18n_ffsec.po"
            },
        ]
    },
}
```

4. Перезапустите сервис `findface-security`.

```
sudo systemctl restart findface-security.service
```

Новый язык будет автоматически применен к системе и отныне доступен на вкладке *Настройки* -> *Внешний вид*. Вы можете в любой момент переключиться между ним и другими доступными языками.

См.также:

Язык веб-интерфейса

2.8.4 Использование нескольких видеокарт

Если на физическом сервере установлено несколько видеокарт, вы можете создать дополнительные экземпляры `findface-extract-api-gpu` или `findface-video-worker-gpu` и распределить их по одному экземпляру на карту.

В этом разделе:

- *Распределение экземпляров `findface-extraction-api-gpu` по видеокартам*
- *Привязка `findface-video-worker-gpu` к дополнительной видеокарте*

Распределение экземпляров `findface-extraction-api-gpu` по видеокартам

Для распределения экземпляров `findface-extraction-api-gpu` по нескольким видеокартам выполните следующие действия:

1. Подготовьте исходный файл конфигурации `findface-extraction-api-gpu` для последующего копирования. Откройте `/etc/findface-extraction-api.ini`. Ограничьте количество экземпляров `findface-extraction-api-gpu` до 1 на видеокарту.

```
sudo vi /etc/findface-extraction-api.ini

...
extractors:
    instances: 1
```

Совет: При необходимости внесите другие коррективы. Например, можно включить распознавание дополнительных атрибутов. См. *Распознавание атрибутов лица*, *Распознавание автомобилей и их атрибутов*, *Распознавание силуэтов и их атрибутов*.

2. Создайте несколько копий файла конфигурации `/etc/findface-extraction-api.ini`, в зависимости от того, какое количество видеокарт будет использоваться для извлечения векторов признаков. Добавьте номера устройств GPU, на которых будут запущены экземпляры, к именам новых файлов конфигурации (устройства GPU №0 и №7 в примере ниже).

Примечание: Нумерация устройств GPU в системе по умолчанию начинается с №0. Чтобы получить список используемых видеокарт, выполните следующую команду:

```
nvidia-smi
```

```
cp findface-extraction-api.ini findface-extraction-api-0.ini
cp findface-extraction-api.ini findface-extraction-api-7.ini
```

- Откройте новые файлы конфигурации. Укажите номера соответствующих устройств GPU и измените номера слушающих портов. Каждый экземпляр должен слушать на своем порту.

```
sudo vi /etc/findface-extraction-api-0.ini

listen: 127.0.0.1:18667
...

gpu_device: 0
...
```

```
sudo vi /etc/findface-extraction-api-7.ini

listen: 127.0.0.1:18668
...

gpu_device: 7
...
```

- Чтобы экземпляры `findface-extraction-api-gpu` работали в одной системе, объедините их через балансировщик нагрузки, например, `nginx`. Для простоты мы рекомендуем настроить работу `nginx` на стандартном порту `findface-extraction-api` 18666 (подробнее см. ниже).

Для настройки балансировки нагрузки выполните следующие действия:

- Создайте новый файл конфигурации `nginx`.

```
sudo vi /etc/nginx/sites-available/lb_extractions
```

- Вставьте текст следующего содержания в только что созданный файл. Обязательно укажите в директиве `server` фактические порты, на которых слушают экземпляры `findface-extraction-api-gpu` (в нашем примере 18667, 18668).

```
upstream extractions {
    server 127.0.0.1:18667 max_fails=3 fail_timeout=30s;
    server 127.0.0.1:18668 max_fails=3 fail_timeout=30s;
}

server {
    listen 18666 default_server;

    server_name _;

    location / {
        client_max_body_size 100m;
        proxy_pass http://extractions;
    }
}
```

3. Включите балансировщик нагрузки.

```
ln -s /etc/nginx/sites-available/lb_extractions /etc/nginx/sites-enabled/lb_
↪extractions
nginx -t
nginx -s reload
```

5. Остановите исходный сервис `findface-extraction-api-gpu` и удалите его из автозагрузки Ubuntu.

```
systemctl stop findface-extraction-api.service
systemctl disable findface-extraction-api.service
```

6. Создайте новый сервис для экземпляров `findface-extraction-api-gpu`.

```
/etc/systemd/system/findface-extraction-api@.service
```

Вставьте следующий текст:

```
[Unit]
Description=Findface Extraction API service %i instance

[Service]
User=ntech
Restart=always
RestartSec=5
Type=notify
ExecStart=/usr/bin/findface-extraction-api -config="/etc/findface-extraction-api-%i.
↪ini"
TimeoutStartSec=60m

[Install]
WantedBy=multi-user.target
```

7. Запустите экземпляры `findface-extraction-api-gpu`. Первый запуск может занять до 10 минут.

```
systemctl daemon-reload
systemctl enable findface-extraction-api@{0,7}
systemctl start findface-extraction-api@{0,7}
```

Совет: Для проверки статуса выполните следующую команду:

```
sudo systemctl list-units 'findface*' -a
```

Привязка findface-video-worker-gpu к дополнительной видеокарте

Для создания дополнительного экземпляра `findface-video-worker-gpu` и его привязки к свободной видеокарте выполните следующие действия:

1. Отобразите статус исходного сервиса `findface-video-worker-gpu`, выполнив команду:

```
sudo systemctl status findface-video-worker-gpu.service
```

2. Найдите полный путь к сервису в следующей строке:

```
Loaded: loaded (/usr/lib/systemd/system/findface-video-worker-gpu.service); enabled;
↪ vendor preset: enabled
```

В приведенном примере это `findface-video-worker-gpu.service` (имя может быть другим). Создайте копию сервиса под новым именем.

```
sudo cp /usr/lib/systemd/system/findface-video-worker-gpu.service /usr/lib/systemd/
↪system/findface-video-worker-gpu2.service`
```

3. Таким же образом создайте под новым именем копию исходного файла конфигурации.

```
sudo cp /etc/findface-video-worker-gpu.ini /etc/findface-video-worker-gpu2.ini
```

4. Откройте только что созданный файл конфигурации и актуализируйте номер используемого GPU-устройства. Измените номер порта в секции `streamer` по следующей формуле: 18999 (номер порта для GPU 0) - номер GPU-устройства, т. е. для GPU 1, `port = 18998`, для GPU 2, `port = 18997` и т. д.

```
sudo vi /etc/findface-video-worker-gpu2.ini

## cuda device number
device_number = 1

...

#-----
[streamer]
#-----
## streamer/shots webserver port, 0=disabled
## type:number env:CFG_STREAMER_PORT longopt:--streamer-port
port = 18999
...
```

5. Откройте новый сервис и укажите только что созданный файл конфигурации.

```
sudo vi /usr/lib/systemd/system/findface-video-worker-gpu2.service

ExecStart=/usr/bin/findface-video-worker-gpu --config /etc/findface-video-worker-
↪gpu2.ini
```

6. Для применения изменений перезагрузите демон `systemd`.

```
sudo systemctl daemon-reload
```

7. Добавьте новый сервис в автозагрузку.

```
sudo systemctl enable findface-video-worker-gpu2.service

Created symlink from /etc/systemd/system/multi-user.target.wants/findface-video-
worker-gpu2.service to /usr/lib/systemd/system/findface-video-worker-gpu2.service
```

8. Запустите новый сервис.

```
sudo systemctl start findface-video-worker-gpu2.service
```

9. Проверьте статус обоих сервисов findface-video-worker-gpu.

```
sudo systemctl status findface-video-worker-* | grep -i 'Active:' -B 3

findface-video-worker-gpu2.service - findface-video-worker-gpu daemon
  Loaded: loaded (/usr/lib/systemd/system/findface-video-worker-gpu2.service;
  enabled; vendor preset: enabled)
  Active: active (running) since Thu 2019-07-18 10:32:02 MSK; 1min 11s ago
...

findface-video-worker-gpu.service - findface-video-worker-gpu daemon
  Loaded: loaded (/usr/lib/systemd/system/findface-video-worker-gpu.service;
  enabled; vendor preset: enabled)
  Active: active (running) since Mon 2019-07-15 15:18:33 MSK; 2 days ago
```

2.8.5 Миграция векторов признаков на другую модель

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/findface_dump`.

Совет: Подробнее см. *Резервное копирование и восстановление FindFace Multi и всех данных*.

```
sudo mkdir -p /etc/findface_dump
sudo cd /etc/findface_dump
sudo findface-storage-api-dump -config /etc/findface-sf-api.ini
```

2. Остановите сервис findface-sf-api.

```
sudo systemctl stop findface-sf-api.service
```

3. Создайте новые шарды, в которых будут храниться заново сгенерированные векторы признаков.
 1. Откройте директорию `/etc/tarantool/instances.available/` и подсчитайте количество шардов по количеству файлов конфигурации `shard-*.lua`.

Примечание: В примере ниже четыре шарда.

```
cd /etc/tarantool/instances.available/

ls -l

shard-001.lua
shard-002.lua
shard-003.lua
shard-004.lua
```

2. Создайте новые шарды в том же количестве, скопировав файлы конфигурации `shard-*.lua`.

Примечание: Для удобства в качестве второй цифры в новых именах используется 1: `shard-01*.lua`.

```
sudo cp shard-001.lua shard-011.lua
sudo cp shard-002.lua shard-012.lua
sudo cp shard-003.lua shard-013.lua
sudo cp shard-004.lua shard-014.lua
```

3. В файле конфигурации каждого шарда измените следующие строки в зависимости от его имени (в нашем примере `shard-011`, `shard-012` и т. д.):

Прежнее значение		Новое значение
<code>listen = „127.0.0.1:32001“</code>		<code>Listen = „127.0.0.1:32011“</code>
<code>vinyl_dir</code> <code>„/opt/ntech/var/lib/tarantool/shard-001“</code>	=	<code>vinyl_dir</code> <code>„/opt/ntech/var/lib/tarantool/shard-011“</code>
<code>work_dir</code> <code>„/opt/ntech/var/lib/tarantool/shard-001“</code>	=	<code>work_dir</code> <code>„/opt/ntech/var/lib/tarantool/shard-011“</code>
<code>memtx_dir</code> <code>„/opt/ntech/var/lib/tarantool/shard-001/snapshots“</code>	=	<code>memtx_dir</code> <code>„/opt/ntech/var/lib/tarantool/shard-011/snapshots“</code>
<code>wal_dir</code> <code>„/opt/ntech/var/lib/tarantool/shard-001/xlogs“</code>	=	<code>wal_dir</code> <code>„/opt/ntech/var/lib/tarantool/shard-011/xlogs“</code>
<code>FindFace.start(«127.0.0.1», 8101, {</code>		<code>FindFace.start(«127.0.0.1», 8111, {</code>

4. Создайте символические ссылки на новые шарды.

```
cd /etc/tarantool/instances.enabled/

sudo ln -s /etc/tarantool/instances.available/shard-01*.lua /etc/tarantool/
↪instances.enabled/
```

5. Создайте директории, в которых будут храниться файлы новых шардов. Назначьте права на созданные директории.

```
cd /opt/ntech/var/lib/tarantool/

mkdir -p shard-01{1..4}/{index,snapshots,xlogs}

chown tarantool:tarantool shard-01* shard-01*/*
```

4. Откройте файл конфигурации `/etc/findface-extraction-api` и замените модели `extraction` на новые в параметрах `body_emben`, `car_emben` и `face_emben`, в зависимости от мигрируемых типов объектов.

```
sudo vi /etc/findface-extraction-api.ini

extractors:
...
models:
...
body_emben: pedrec/<new_model_body>.gpu.fnk
...
car_emben: carrec/<new_model_car>.gpu.fnk
...
face_emben: face/<new_model_face>.gpu.fnk
```

Перезапустите сервис `findface-extraction-api`.

```
sudo systemctl restart findface-extraction-api.service
```

5. Запустите новые шарды.

```
for i in {11..14}; do sudo systemctl start tarantool@shard-0$i; done
```

6. На основании приведенного ниже примера создайте файл конфигурации с настройками миграции `<migration.ini>`.

```
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
```

(continues on next page)

(продолжение с предыдущей страницы)

```

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: ""
storage-api-to:
  timeouts:
    connect: 5s
    response_header: 30s
    overall: 35s
    idle_connection: 10s
max-idle-conns-per-host: 20
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: ""
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/

```

(continues on next page)

(продолжение с предыдущей страницы)

```

slave: ""
- master: http://127.0.0.1:8103/v2/
  slave: ""
- master: http://127.0.0.1:8104/v2/
...

```

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: ""
...

```

7. Запустите утилиту `findface-sf-api-migrate` с опцией `-config` и укажите файл конфигурации `<migration.ini>`.

```
findface-sf-api-migrate -config migration.ini
```

Примечание: Если в системе много событий и карточек, процесс миграции может занять значительное количество времени.

8. После завершения миграции остановите шарды и отключите их автоматический запуск в ОС (не удаляйте их).

```

for i in {01..04}; do sudo systemctl stop tarantool@shard-0$i.service ; done

for i in {01..04}; do sudo systemctl disable tarantool@shard-0$i.service ; done

```

9. Откройте файл конфигурации `/etc/findface-sf-api.ini` и измените порты шардов в зависимости от новых настроек. Перезапустите сервис `findface-sf-api`.

```

sudo vi /etc/findface-sf-api.ini

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: ""

```

(continues on next page)

(продолжение с предыдущей страницы)

```
sudo systemctl start findface-sf-api.service
```

10. Импортируйте структуру базы данных из файла `tnt_schema.lua`.

```
sudo findface-security make_tnt_schema | sudo tee /etc/findface-security/tnt_schema.  
↪lua
```

См.также:

Изменение структуры базы данных векторов признаков.

11. Перенесите схему базы данных из FindFace Multi в PostgreSQL, заново создайте *предустановленные* пользовательские роли и первого администратора.

```
sudo findface-security migrate  
sudo findface-security create_groups  
sudo findface-security create_default_user
```

12. Перезагрузите сервисы.

```
sudo systemctl restart findface-security.service  
sudo systemctl restart findface-extraction-api findface-video-worker* findface-  
↪video-manager findface-sf-api
```

13. 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 findface-security migrate_clusters --face --body --car --use-best-event --use-  
↪thumbnail --force-clusterin
```

В результате система повторно сгенерирует векторы признаков для существующих кластерных событий и автоматически запустит запланированную кластеризацию для пересборки кластеров.

2.8.6 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 объектов 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 `/etc/findface-security/tnt_schema.lua`.

Структура представляет собой набор полей, каждое из которых описывается следующими параметрами:

- `id`: id поля;
- `name`: название поля, должно совпадать с названием соответствующего параметра объекта;
- `field_type`: тип данных;
- `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. Остановите сервис `findface-security`.

```
sudo systemctl stop findface-security.service
```

2. Создайте резервную копию базы данных векторов признаков в любой выбранной директории, например, `/etc/findface_dump`.

Совет: Подробнее см. *Резервное копирование и восстановление FindFace Multi и всех данных*.

```
mkdir -p /etc/findface_dump
cd /etc/findface_dump
sudo findface-storage-api-dump -config /etc/findface-sf-api.ini
```

3. Подготовьте файл `tnt_schema.lua` с новой структурой базы данных.
4. Измените структуру базы данных с помощью подготовленного файла `tnt_schema.lua`.

```
sudo findface-security make_tnt_schema | sudo tee /etc/findface-security/tnt_schema.
↪lua
```

5. Перейдите в директорию с файлами конфигурации Tarantool `/etc/tarantool/instances.available/`. Убедитесь, что перед секцией `FindFace.start` добавлена строка `dofile("/etc/findface-security/tnt_schema.lua")`, а переменные `meta_scheme` и `meta_indexes` определены в параметрах `FindFace.start`.

```
sudo vi /etc/tarantool/instances.available/<shard_00N>.lua

dofile("/etc/findface-security/tnt_schema.lua")

FindFace.start("127.0.0.1", 8101, {
    license_ntls_server="127.0.0.1:3133",
    meta_indexes=meta_indexes,
    meta_scheme = meta_scheme
})
```

6. Остановите шарды findface-tarantool-server. Удалите данные из каталогов, соответствующих активным шардам.

```
sudo systemctl stop 'tarantool@*'

sudo rm /opt/ntech/var/lib/tarantool/shard-*/{index,snapshots,xlogs}/*
```

7. 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
```

8. Восстановите базу данных Tarantool из резервной копии.

Важно: Если некоторые прежние поля отсутствуют в новой структуре базы данных, сначала потребуется вручную удалить соответствующие данные из резервной копии.

```
cd /etc/findface_dump
for x in *.json; do curl -X POST "http://127.0.0.1:18411/v2/galleries/${x%%.json}";
done
for x in *.json; do sudo findface-storage-api-restore -config /etc/findface-sf-api.
ini < "$x"; done
```

9. Запустите сервис findface-security.

```
sudo systemctl start findface-security.service
```

См.также:

Пользовательские метаданные в Tarantool

2.8.7 Удаление экземпляра продукта

Вы можете автоматически удалить FindFace Multi вместе с используемыми *хранилищами данных*, используя скрипт `ffmulti_uninstall.sh`. Перед удалением по запросу будут созданы резервные копии файлов конфигурации и хранилищ данных.

Важно: Что касается директорий с *артефактами FindFace Multi*, скрипт создаст резервные копии и очистит только директории `/var/lib/findface-security/uploads` и `/var/lib/ffupload/` (заданы по умолчанию). Если вы изменили место хранения артефактов, вам придется создать резервную копию

и очистить его самостоятельно. См. *Резервное копирование и восстановление FindFace Multi и всех данных* для справки.

Выполните следующие действия:

1. Загрузите скрипт `ffmulti_uninstall.sh` в любой каталог на сервере установки (например, в `/home/username/`).
2. Из данного каталога сделайте скрипт исполняемым.

```
sudo chmod +x ffmulti_uninstall.sh
```

3. Запустите скрипт.

```
sudo ./ffmulti_uninstall.sh
```

4. На вопрос скрипта о выборе режима удаления ответьте **backup**, чтобы создать резервную копию, а затем удалить FindFace Multi вместе с хранилищами данных. Ответьте **all**, чтобы полностью стереть FindFace Multi и хранилища данных без резервного копирования.
5. Скрипт удаляет все содержимое директории `/var/lib/ffupload`. Однако сама директория остается в файловой системе. Удалите `/var/lib/ffupload` вручную, выполнив следующую команду:

```
sudo rmdir /var/lib/ffupload
```

Важно: Для восстановления FindFace Multi из резервной копии см. *Резервное копирование и восстановление FindFace Multi и всех данных*.

2.8.8 Проверка статуса компонентов

Проверьте статус компонентов, если вы столкнулись с проблемой в системе.

Компонент	Команда для просмотра статуса сервиса
<code>findface-extraction-api</code>	<code>sudo systemctl status findface-extraction-api.service</code>
<code>findface-sf-api</code>	<code>sudo systemctl status findface-sf-api.service</code>
<code>findface-tarantool-server</code>	<code>sudo systemctl status tarantool.service</code>
Шарды <code>findface-tarantool-server</code>	<code>sudo systemctl status tarantool@shard-00*</code>
<code>findface-video-manager</code>	<code>sudo systemctl status findface-video-manager.service</code>
<code>findface-video-worker</code>	<code>sudo systemctl status findface-video-worker*.service</code>
<code>findface-ntls</code>	<code>sudo systemctl status findface-ntls</code>
<code>findface-security</code>	<code>sudo systemctl status findface-security.service</code>
<code>findface-counter</code>	<code>sudo systemctl status findface-counter.service</code>
<code>findface-liveness-api</code>	<code>sudo systemctl status findface-liveness-api.service</code>
<code>etcd</code>	<code>sudo systemctl status etcd.service</code>
<code>NginX</code>	<code>sudo systemctl status nginx.service</code>
<code>memcached</code>	<code>sudo systemctl status memcached.service</code>
<code>postgresql</code>	<code>sudo systemctl status postgresql*</code>
<code>nats</code>	<code>sudo systemctl status nats.service</code>
<code>pgbouncer</code>	<code>sudo systemctl status pgbouncer.service</code>

2.8.9 Логи сервисов

При разборе нештатных ситуаций используйте логи сервисов FindFace Multi, содержащие подробную детализировку всех событий, произошедших в системе.

В этом разделе:

- *Настройка логирования*
- *Просмотр логов сервисов*

Настройка логирования

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**.

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
```

2. 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 K, M, G, T, P, E в качестве единиц для указанного размера (соответственно 1024, 1024², ... байт).

```
...
SystemMaxUse=3G
```

3. Перезапустите сервис **journald**.

```
sudo systemctl restart systemd-journald.service
```

4. Остановите и отключите сервис **syslog**.

```
sudo systemctl stop syslog.socket rsyslog.service
sudo systemctl disable syslog.socket rsyslog.service
```

5. При необходимости удалите существующие файлы логов, созданные посредством `syslog`, а также логи ядра.

```
sudo rm /var/log/syslog*
sudo rm /var/log/kern.log*
```

Просмотр логов сервисов

Для того посмотреть логи для каждого компонента, используя команду `journalctl -u <component>`, например:

```
journalctl -u findface-extraction-api
```

См.также:

Журнал действий пользователей

2.8.10 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 в 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 Multi 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
    },
    {
      "type": "boolean",
      "name": "emotions",
```

(continues on next page)

```
    "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"
  }
]
}
```

2.8.11 Автоматическое восстановление Tarantool

Важно: Мы настоятельно рекомендуем вам правильно выключать серверы Tarantool в каждой нестабильной ситуации. Данная мера предотвратит повреждение данных, поэтому функционал, описанный в настоящем разделе, будет не нужен.

Предупреждение: Будьте предельно осторожны с включением данной функции, так как это может привести к молчаливой потере данных.

Если архитектура вашей системы не обеспечивает бесперебойную доступность серверов Tarantool, рекомендуется включить автоматическое восстановление базы данных. В этом случае каждый раз при возникновении ошибки во время чтения файла `.snap` или `.xlog`, Tarantool попытается прочитать как можно больше информации и восстановить файл, игнорируя битые записи.

Для включения автоматического восстановления базы данных выполните следующие действия:

1. Для каждого шарда Tarantool откройте файл конфигурации `/etc/tarantool/instances.available/shard-*.lua` и раскомментируйте строку `force_recovery = true`.

```
sudo vi /etc/tarantool/instances.available/shard-*.lua

box.cfg{
    force_recovery = true,
}
```

2. Перезапустите шарды.

```
systemctl restart tarantool@shard-*
```

2.8.12 Ручная очистка базы данных от старых данных

Совет: Для настройки автоматического удаления старых данных см. *Автоматическое удаление старых событий и эпизодов*.

Для ручного удаления старых данных из базы данных FindFace Multi используйте утилиту `cleanup`. Вы можете выборочно удалить следующие данные:

- события без совпадений (лица, силуэты, автомобили) и связанные эпизоды,
- события без совпадений (лица, силуэты, автомобили) и связанные эпизоды,
- полные кадры событий с совпадениями (лица, силуэты, автомобили),
- полные кадры событий без совпадений (лица, силуэты, автомобили),
- записи счетчика,
- кластерные события (лица, силуэты, автомобили),
- журнал действий пользователей,

- активации областей.

Справка по утилите `cleanup` вызывается следующей командой:

```
sudo findface-security cleanup --help
usage: findface-security 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]
                                [--area-activations-max-age AREA_ACTIVATIONS_MAX_AGE]
                                [--audit-logs-max-age AUDIT_LOGS_MAX_AGE]
                                [--counter-records-max-age COUNTER_RECORDS_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 days)
--face-events-max-fullframe-matched-age FACE_EVENTS_MAX_FULLFRAME_MATCHED_AGE
```

(continues on next page)

(продолжение с предыдущей страницы)

```

        face events max fullframe matched age to clean up (in
        days)
--face-events-max-unmatched-age FACE_EVENTS_MAX_UNMATCHED_AGE
        face events max unmatched age to clean up (in days)
--face-events-max-matched-age FACE_EVENTS_MAX_MATCHED_AGE
        face events max matched age to clean up (in days)
--body-events-max-fullframe-unmatched-age BODY_EVENTS_MAX_FULLFRAME_UNMATCHED_AGE
        body events max fullframe unmatched age to clean up
        (in days)
--body-events-max-fullframe-matched-age BODY_EVENTS_MAX_FULLFRAME_MATCHED_AGE
        body events max fullframe matched age to clean up (in
        days)
--body-events-max-unmatched-age BODY_EVENTS_MAX_UNMATCHED_AGE
        body events max unmatched age to clean up (in days)
--body-events-max-matched-age BODY_EVENTS_MAX_MATCHED_AGE
        body events max matched age to clean up (in days)
--car-events-max-fullframe-unmatched-age CAR_EVENTS_MAX_FULLFRAME_UNMATCHED_AGE
        car events max fullframe unmatched age to clean up (in
        days)
--car-events-max-fullframe-matched-age CAR_EVENTS_MAX_FULLFRAME_MATCHED_AGE
        car events max fullframe matched age to clean up (in
        days)
--car-events-max-unmatched-age CAR_EVENTS_MAX_UNMATCHED_AGE
        car events max unmatched age to clean up (in days)
--car-events-max-matched-age CAR_EVENTS_MAX_MATCHED_AGE
        car events max matched age to clean up (in days)
--car-cluster-events-max-age CAR_CLUSTER_EVENTS_MAX_AGE
        car cluster events max age to clean up (in days)
--body-cluster-events-max-age BODY_CLUSTER_EVENTS_MAX_AGE
        body cluster events max age to clean up (in days)
--face-cluster-events-max-age FACE_CLUSTER_EVENTS_MAX_AGE
        face cluster events max age to clean up (in days)
--area-activations-max-age AREA_ACTIVATIONS_MAX_AGE
        area activations max age to clean up (in days)
--audit-logs-max-age AUDIT_LOGS_MAX_AGE
        audit logs max age to clean up (in days)
--counter-records-max-age COUNTER_RECORDS_MAX_AGE
        counter records max age to clean up (in days)
--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

```

(continues on next page)

(продолжение с предыдущей страницы)

	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 выполните команду:

```
sudo findface-security cleanup --car-events-max-unmatched-age 5
```

Для удаления событий распознавания автомобилей с совпадениями старше 5 дней выполните команду:

```
sudo findface-security cleanup --car-events-max-matched-age 5
```

Следующие команды удаляют только полные кадры событий распознавания силуэтов с совпадениями и без:

```
sudo findface-security cleanup --body-events-max-fullframe-matched-age 5
sudo findface-security cleanup --body-events-max-fullframe-unmatched-age 5
```

Для того чтобы удалить только записи счетчика, выполните команду:

```
sudo findface-security cleanup --counter-records-max-age 5
```

Для того чтобы удалить только кластерные события с лицами, выполните команду:

```
sudo findface-security cleanup --face-cluster-events-max-age 5
```

Для того чтобы удалить только записи в журнале действий пользователей, выполните команду:

```
sudo findface-security cleanup --audit-logs-max-age 5
```

Для того чтобы удалить только активации областей, выполните команду:

```
sudo findface-security cleanup --area-activations-max-age 5
```

Важно: Должен быть задан хотя бы один аргумент из указанных.

2.8.13 Отключение сервисов

Существует возможность отключить следующие сервисы FindFace Multi, если вы ими не пользуетесь:

- эпизоды
- менеджер очередей обработки видеоархивов
- вебхуки

Для этого откройте файл конфигурации `/etc/findface-security/config.py` и внесите изменения в раздел `SERVICES`, установив значение `False` для тех сервисов, которые нужно отключить. Перезапустите `findface-security`.

```

sudo vi /etc/findface-security/config.py

# disable unused services to increase
# overall system performance in some cases.
SERVICES = {
    "ffsecurity": {
        "episodes": True,
        "webhooks": True,
        # use queue manager to prevent drops of video archive events
        "video_archive_events_manager": True,
    }
}

```

```
sudo systemctl restart findface-security.service
```

В результате соответствующие вкладки исчезнут из веб-интерфейса.

Примечание: Вкладка останется, если на ней есть какие-либо объекты (например, на вкладке *Веб-хуки* созданы вебхуки). Однако сервис перестанет работать и генерировать данные.

2.8.14 Скрытие пунктов меню

Для того чтобы скрыть определенные пункты меню, выполните следующие действия:

1. Откройте файл конфигурации `/etc/findface-security/config.py`.

```
sudo vi /etc/findface-security/config.py
```

2. В раздел `FFSECURITY_UI_CONFIG` вставьте секцию `menu`, как показано в примере ниже. Перечислите пункты меню, которые вы хотите скрыть.

```

FFSECURITY_UI_CONFIG = {
    "menu": {
        "disabled_items": ["video-wall", "know_your_customer"]
    },
}

```

Вы можете скрыть следующие пункты:

Пункт меню	Настройка
<i>Поиск</i>	"search"
<i>События</i>	"events"
<i>Эпизоды</i>	"episodes"
<i>Кластеры</i>	"clusters"
<i>Анализ взаимодействий</i>	"contacts"
<i>Счетчики</i>	"counters"
<i>Камеры</i>	"cameras"
<i>Видеостена</i>	"video-wall"
<i>Отчеты</i>	"reports"
<i>Журнал действий</i>	"audit_logs"
<i>Карточки</i>	"cards"
<i>Видео</i>	"videos"
<i>Аналитика</i>	"know_your_customer"
<i>Настройки</i>	"preferences"
<i>Основные настройки</i>	"settings"
<i>Внешний вид</i>	"appearance"
<i>Роли</i>	"roles"
<i>Пользователи</i>	"users"
<i>Группы камер</i>	"camera-groups"
<i>Списки наблюдения</i>	"watch-lists"
<i>Лицензия</i>	"license"
<i>Инструменты</i>	"tools"
<i>Сравнение</i>	"verify"
<i>Разработчикам</i>	"developer"
<i>Документация</i>	"documentation"
<i>API-документация</i>	"api_doc"
<i>Вебхуки</i>	"webhooks"
пункты меню, появляющиеся при активации специализированных плагинов	Свяжитесь с нашей технической поддержкой, чтобы узнать подробнее о плагинах в своей системе.

3. Перезапустите сервис findface-security.

```
sudo systemctl restart findface-security.service
```

2.8.15 Сброс пароля

Для того чтобы сбросить пользовательский пароль в веб-интерфейсе FindFace Multi, выполните следующую команду:

```
findface-security changepassword %username
```


2.8.16 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
- 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
sudo chown ntech:ntech /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

Для миграции хранилища фотографий FindFace Multi выполните следующие действия:

1. Остановите сервис `findface-security`, чтобы предотвратить потерю данных.

```
sudo systemctl stop findface-security
```

2. По умолчанию фотографии хранятся в директории `/var/lib/`. Выполните миграцию хранилища на *новый диск*.

```
sudo cp -ax /var/lib/findface-security/ -R /mnt/ffdata/  
sudo rm -r /var/lib/findface-security/  
sudo cp -ax /var/lib/ffupload/ -R /mnt/ffdata/  
sudo rm -r /var/lib/ffupload/
```

3. Создайте символические ссылки на новые директории.

```
sudo ln -s /mnt/ffdata/findface-security/ /var/lib/  
sudo ln -s /mnt/ffdata/ffupload/ /var/lib/
```

4. Убедитесь, что права назначены верно.

```
sudo chown ntech:ntech /mnt/ffdata/findface-security/
```

5. Запустите сервис `findface-security`.

```
sudo systemctl start findface-security
```

Миграция основной базы данных (PostgreSQL)

Для того чтобы мигрировать базу данных PostgreSQL, выполните следующие действия:

1. Остановите сервисы `findface-security`, `pgbouncer` и PostgreSQL.

```
sudo systemctl stop findface-security.service pgbouncer.service postgresql.service  
↪ postgresql@10-main.service
```

2. Создайте директорию для базы данных на *новом диске*.

```
sudo mkdir -p /mnt/ffdata/some_directory/db
```

3. Перенесите базу данных в новую директорию.

```
sudo mv /var/lib/postgresql /mnt/ffdata/some_directory/db
```

4. Создайте символическую ссылку на новую директорию.

```
sudo ln -s /mnt/ffdata/some_directory/db/postgresql /var/lib/postgresql
```

5. Запустите сервисы PostgreSQL, `pgbouncer` и `findface-security`.

```
sudo systemctl start postgresql.service postgresql@10-main.service pgbouncer.service  
sudo systemctl start findface-security.service
```

2.8.17 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 сервис `findface-liveness-api` и отключите его автозагрузку, выполнив следующую последовательность команд:

```
sudo systemctl stop findface-liveness-api.service && sudo systemctl disable
↪ findface-liveness-api.service
```

2. Откройте файл конфигурации `/etc/findface-extraction-api.ini`.

```
sudo vi /etc/findface-extraction-api.ini
```

3. Отключите модель нейронной сети, используемую сервисом `findface-liveness-api`, передав пустое значение `''` в параметр `face_liveness`.

Примечание: Не удаляйте сам параметр, поскольку в этом случае будет выполняться поиск модели по умолчанию.

```
extractors:
  models:
    ...
    face_liveness: ''
    ...
```

4. Перезапустите `findface-extraction-api`.

```
sudo systemctl restart findface-extraction-api
```

2.8.18 Устранение неполадок с дистрибутивом MongoDB

Совет: По вопросам устранения неполадок обращайтесь к нашим специалистам по адресу support@ntechlab.com.

Для работы функционала Видеомэгнитофон требуется стороннее программное обеспечение MongoDB.

Примечание: MongoDB – документно-ориентированная СУБД.

В настоящий момент на территории России и Беларуси установка MongoDB из публичного репозитория невозможна. Если вы находитесь на территории данных стран и намерены использовать функционал Видеомэгнитофон, вам необходимо вручную установить базу данных MongoDB из deb-пакета перед проведением любой процедуры развертывания FindFace Multi.

Выполните следующие действия:

1. Установите предоставленный технической поддержкой дистрибутив MongoDB с помощью команды:

```
sudo dpkg -i <mongodb-repo-bionic-4.4_2022.06.10_amd64>.deb
```

2. Выполните развертывание FindFace Multi с видеомagniофоном по любой из следующих схем:

- При *развертывании из консольного инсталлятора*, на вопрос «Do you want to install NtechLab VMS? (Хотите ли Вы установить NtechLab VMS)» ответить «No (Нет)». Далее выполните *Пошаговое развертывание функционала Видеомagniофон*.
- При *развертывании из консольного инсталлятора*, на вопрос «Do you want to install NtechLab VMS? (Хотите ли Вы установить NtechLab VMS)» ответить «Yes (Да)», чтобы установить Видеомagniофон.
- Выполните *Базовое пошаговое развертывание из apt-репозитория и Пошаговое развертывание функционала Видеомagniофон*.

2.9 Приложения

2.9.1 Подробно о компонентах

`findface-extraction-api`

Сервис `findface-extraction-api` использует нейронные сети для обнаружения объектов на изображении, извлечения векторов признаков объектов, распознавания атрибутов объектов (например, цвета одежды для силуэтов).

Компонент взаимодействует с сервисом `findface-sf-api` следующим образом:

- Получает от него фотографию с объектом или нормализованное изображение объекта.
- Возвращает координаты рамки с объектом и, при наличии запроса от ``findface-sf-api``, вектор признаков объекта и данные об его атрибутах.

Полный список функций:

- детекция (обнаружение) объекта на исходном изображении с возвращением координат рамки с объектом,
- получение из исходного изображения нормализованного изображения объекта,
- извлечение из нормализованного изображения лица вектора признаков (биометрического образца),
- распознавание атрибутов объекта (пол человека, возраст, эмоции; цвет одежды; цвет автомобиля, модель автомобиля и др.).

Сервис `findface-extraction-api` может работать с ускорением на CPU (устанавливается из пакета `findface-extraction-api`) или GPU (устанавливается из пакета `findface-extraction-api-gpu`). Как для CPU-, так и для GPU-сервиса, настройка выполняется через файл конфигурации `/etc/findface-extraction-api.ini`. Вы можете просмотреть его исходное содержимое [здесь для CPU](#), [здесь для GPU](#).

Пользовательская настройка `findface-extraction-api` (как CPU, так и GPU) выполняется с использованием следующих параметров:

Параметр	Описание
<code>cheetah -> min_object_size</code>	Минимальный размер лица, которое будет гарантированно найдено. Определяется размером рамки с лицом (bbox). Чем больше значение, тем менее ресурсоемок процесс обнаружения лица.
<code>gpu_device</code>	(Только для GPU) Номер GPU-устройства, используемого <code>findface-extraction-api-gpu</code> .
<code>license_ntls_server</code>	Адрес и порт сервера лицензирования <code>findface-ntls</code> .

При необходимости вы также можете включить модели для распознавания атрибутов лица, силуэтов и их атрибутов, автомобилей и их атрибутов, модели для распознавания liveness лиц:

- *Распознавание атрибутов лица.*
- *Распознавание живых лиц в реальном времени (Liveness)*
- *Liveness как автономный сервис*
- *Распознавание силуэтов и их атрибутов*
- *Распознавание автомобилей и их атрибутов*

Важно: Тип ускорения модели (CPU/GPU) 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: Для того чтобы отключить модель экстрактора, передайте в соответствующий параметр пустое значение. Не удаляйте сам параметр, поскольку в этом случае будет выполняться поиск модели по умолчанию.

```
...
extractors:
...
  models:
    body_color: ''
    body_emben: ''
    body_quality: ''
    car_color: ''
    car_description: ''
    car_emben: ''
    ...
```

`findface-sf-api`

Сервис `findface-sf-api` реализует HTTP API для доступа к функциям ядра FindFace, таким как обнаружение (детекция) объектов и распознавание объектов.

Примечание: Сами указанные функции выполняются компонентом `findface-extraction-api`.

Компонент `findface-sf-api` взаимодействует со следующими компонентами ядра FindFace:

- база данных векторов признаков на основе Tarantool через сервис `findface-tarantool-server`
- компонент `findface-extraction-api`, обеспечивающий обнаружение объектов и распознавание объектов
- компонент `findface-upload`, отвечающий за хранение исходных изображений и артефактов ядра FindFace core

Для обнаружения объекта на фотографии в компонент `findface-sf-api` должен быть отправлен API-запрос, передающий данную фотографию в виде файла или URL. Данный запрос затем перенаправляется в компонент `findface-extraction-api`.

При наличии в системе видеодетектора объектов (присутствует в FindFace Multi) компонент `findface-sf-api` получает данные об обнаруженных объектах вместе с правилами их обработки от компонента `findface-facerouter` и затем выполняет полученные директивы (например, сохраняет объекты в определенную галерею базы данных).

Примечание: В FindFace Multi функции `findface-facerouter` выполняет сервис `findface-security`.

Полный список функций:

- HTTP API по части обнаружения и распознавания объектов (сами операции выполняются `findface-extraction-api`).
- сохранение объектов в базу данных векторов признаков (через сервис `findface-tarantool-server`),
- сохранение исходных изображений, миниатюр и нормализованных изображений объектов на веб-сервере nginx (через сервис `findface-upload`).
- обеспечение взаимодействия всех компонентов системы.

Настройка компонента `findface-sf-api` выполняется через файл конфигурации `/etc/findface-sf-api.ini`. Вы можете просмотреть его исходное содержимое [здесь](#).

Пользовательская настройка `findface-sf-api` выполняется с использованием следующих параметров:

Параметр	Описание
<code>extraction-api -> extraction-api</code>	IP-адрес и порт сервера <code>findface-extraction-api</code> .
<code>limits -> body-image-length</code>	Максимальный размер в байтах изображения, передаваемого через API-запрос.
<code>normalized-storage -> webdav -> upload_url</code>	Путь в WebDAV nginx, по которому в компонент <code>findface-upload</code> будут отправляться исходные изображения, миниатюры и нормализованные изображения объектов.
<code>storage-api -> shards -> master</code>	IP-адрес физического сервера с мастером шарда <code>findface-tarantool-server</code> .
<code>storage-api -> shards -> slave</code>	IP-адрес физического сервера с репликой шарда <code>findface-tarantool-server</code> .

`findface-tarantool-server`

Сервис `findface-tarantool-server` обеспечивает взаимодействие между сервисом `findface-sf-api` и базой данных векторов признаков на основе Tarantool следующим образом:

Совет: Подробнее см. [официальную документацию Tarantool](#).

- `findface-tarantool-server` получает от `findface-sf-api` данные для записи в базу данных (например, об обнаруженных объектах).
- По запросу от `findface-sf-api` `findface-tarantool-server` выполняет поиск по базе данных и возвращает его результат.

Для увеличения скорости поиска на каждом сервере с базой данных Tarantool могут быть созданы многочисленные сегменты («шарды») `findface-tarantool-server`. Их параллельное функционирование приводит к значительному увеличению производительности (в 70-100 раз).

Полный список функций:

- сохранение объектов в базу векторов признаков,
- выполнение поиска по базе биометрических данных,
- реализация прямых запросов в базу данных Tarantool (см. [Прямые API-запросы к базе данных Tarantool](#)).

Настройка компонента `findface-tarantool-server` выполняется через файл конфигурации `/etc/tarantool/instances.available/<shard-*>.lua`. Вы можете просмотреть его исходное содержимое [здесь](#).

Важно: В среде с несколькими шардами файл конфигурации настраивается отдельно для каждого шарда.

Пользовательская настройка `findface-tarantool-server` выполняется с использованием следующих параметров:

Параметр	Описание
<code>force_recovery</code>	Включает/отключает автоматическое восстановление базы данных Tarantool. Если автоматическое восстановление данных включено (<code>true</code>), каждый раз при возникновении ошибки во время чтения файла <code>.snap</code> или <code>.xlog</code> , Tarantool попытается прочитать как можно больше информации и восстановить файл, игнорируя битые записи.
<code>license_server</code>	URL-адрес сервера лицензирования <code>findface-ntls</code> .
<code>memtx_memory</code>	Максимальный размер оперативной памяти в байтах, в зависимости от количества объектов на шарде. Перед изменением данного параметра обратитесь к нашим экспертам за консультацией по адресу support@ntechlab.com .
<code>meta_scheme</code>	Структура базы данных для хранения результатов распознавания объектов. Представляет собой набор полей, для каждого из которых указываются следующие параметры: <code>id</code> : id поля, <code>name</code> : название поля, должно совпадать с названием соответствующего параметра объекта, <code>field_type</code> : тип данных, <code>default</code> : значение по умолчанию. Если значение по умолчанию для поля больше <code>'le14 - 1'</code> , то его следует записывать в виде строки, т. е. <code>"123123"</code> вместо <code>123123</code> .

Исходная структура базы данных передается из файла `/etc/findface-security/tnt_schema.lua` в параметр `meta_scheme`. Подробнее см. *Изменение структуры базы данных векторов признаков*.

findface-upload

Компонент `findface-upload` представляет собой веб-сервер на базе WebDAV nginx, который используется для хранения исходных изображений, миниатюр и нормализованных изображений объектов (получает их от компонента `findface-sf-api`).

По умолчанию исходные изображения, миниатюры и нормализованные изображения лиц хранятся в каталоге `/var/lib/ffupload/uploads/`.

Компонент `findface-upload` автоматически настраивается при установке. Настройка не предусмотрена.

Видеодетекция объектов: `findface-video-manager` и `findface-video-worker`

Примечание: Компонент `findface-video-worker` поставляется в пакетах с ускорением на CPU (`findface-video-worker-cpu`) и GPU (`findface-video-worker-gpu`).

В этом разделе:

- Функции `findface-video-manager`
- Функции `findface-video-worker`
- Настройка видеодетекции объектов
- Job-задания

Функции `findface-video-manager`

Сервис `findface-video-manager` является частью модуля видеодетекции объектов и используется для непосредственного управления детекцией объектов на видео.

Сервис `findface-video-manager` взаимодействует с `findface-video-worker` следующим образом:

- Обеспечивает `findface-video-worker` настройками и списком видеопотоков для обработки. Для этого он выдает `findface-video-worker` так называемое *job-задание*, задачу на обработку видео, которая содержит параметры конфигурации и сведения о видеопотоке.
- В распределенной системе распределяет видеопотоки (job-задания) по свободным экземплярам `findface-video-worker`.

Примечание: Параметры конфигурации, передаваемые через job-задания, имеют больший приоритет, чем аналогичные параметры в файле конфигурации `/etc/findface-video-manager.conf`.

Для работы `findface-video-manager` требуется установленный сервис ETCD. ETCD представляет собой стороннее программное обеспечение, реализующее распределенное хранилище ключей `findface-video-manager`. Используется в качестве координационной службы в распределенной системе, обеспечивая отказоустойчивость работы видеодетектора объектов.

Полный список функций `findface-video-manager`:

- конфигурирование параметров видеодетектора объектов
- управление списком видеопотоков для обработки

Функции `findface-video-worker`

Компонент `findface-video-worker` (или `findface-video-worker-gpu`) является частью модуля видеодетекции объектов и служит для обнаружения объектов «на лету» в видеопотоке или видеофайле. Он поддерживает большинство видеоформатов и кодеков, которые могут быть декодированы FFmpeg.

Сервис `findface-video-worker` взаимодействует с сервисами `findface-video-manager` и `findface-facerouter` следующим образом:

- По запросу `findface-video-worker` получает от `findface-video-manager` job-задание с настройками и списком видеопотоков для обработки.
- Сервис `findface-video-worker` отправляет полученные нормализованные изображения объектов вместе с полными кадрами и метаданными, такими как рамка вокруг объекта, ID камеры и время детекции, в сервис `findface-facerouter` для дальнейшей обработки.

Примечание: В FindFace Multi функции `findface-facerouter` выполняются сервисом `findface-security`.

Полный список функций `findface-video-manager`:

- обнаружение объектов на видео,
- извлечение нормализованных изображений объектов,
- поиск наилучшего изображения объектов,
- дедупликация кадров с объектом (только один кадр на каждое событие распознавания объекта).

При обработке видео `findface-video-worker` последовательно использует следующие алгоритмы:

- **Детектор движения.** Данный алгоритм позволяет снизить потребление ресурсов, поскольку детектор объектов включается только по движению в кадре.
- **Детектор объектов.** Детектор объектов отслеживает, обнаруживает и захватывает объекты на видео. Может работать одновременно с несколькими объектами в кадре. С помощью встроенной нейронной сети выполняет поиск кадра с лучшим изображением объекта. Как только лучшее изображение найдено, отправляет его в компонент `findface-facerouter`.

Подбор лучшего изображения объекта может быть выполнен в одном из следующих режимов:

- Режим реального времени
- Буферный режим

Режим реального времени

В режиме реального времени `findface-video-worker` начинает отправлять изображения объекта в компонент `findface-facerouter` на лету после появления объекта в поле зрения видеокамеры. Доступны следующие опции отправки:

- Если параметр `realtime_post_every_interval: true`, детектор объектов выбирает лучший кадр в течение каждого из последовательных промежутков времени, равных `realtime_post_interval`, и отправляет его в `findface-facerouter`.
- Если `realtime_post_every_interval: false`, детектор объектов выбирает лучшее изображение объекта динамически:
 1. Сначала оценивается качество изображения объекта. Если оно превышает некое предустановленное пороговое значение, то объект отправляется в `findface-facerouter`.
 2. Порог повышается после каждой отправки изображения объекта в `findface-facerouter`. Каждый раз, когда детектор объектов получает изображение того же объекта лучшего качества, оно отправляется.
 3. При исчезновении объекта из поля зрения видеокамеры снова устанавливается пороговое значение по умолчанию.
- Если `realtime_post_first_immediately: true`: детектор объектов отправляет первое изображение объекта из трека сразу после того, как он пройдет через фильтры качества, размера и региона захвата (ROI), не дожидаясь окончания первого `realtime_post_interval`. Способ отправки последующих изображений объекта зависит от установленного значения `realtime_post_every_interval`. Если `realtime_post_first_immediately: false`, детектор объектов отправляет первое изображение объекта только после окончания первого `realtime_post_interval`.

Буферный режим

Буферный режим требует меньший объем дискового пространства по сравнению с режимом реального времени, поскольку для каждого объекта компонент `findface-video-worker` отправляет только одно изображение из трека, но наивысшего качества. В этом режиме детектор объектов хранит в буфере видеопоток с объектом до тех пор, пока объект не исчезнет из поля зрения камеры. После этого детектор объектов выбирает лучшее изображение объекта из буферизованного видео и отправляет его в `findface-facerouter`.

Настройка видеодетекции объектов

Настройка видеодетектора объектов выполняется через следующие файлы конфигурации:

1. Файл конфигурации компонента `findface-video-manager` `/etc/findface-video-manager.conf`.
Вы можете просмотреть его исходное содержимое [здесь](#)

Пользовательская настройка `findface-video-manager` выполняется с использованием следующих параметров:

Опция	Описание
etcd endpoints ->	IP-адрес и порт сервиса etcd. Значение по умолчанию: 127.0.0.1:2379.
ntls enabled ->	Если true, компонент findface-video-manager отправляет в компонент findface-video-worker задания только на обработку того количества видеокамер, которое указано в лицензии. Значение по умолчанию: false.
ntls -> url	IP-адрес и порт сервера findface-ntls. Значение по умолчанию: http://127.0.0.1:3185/.
router_url	IP-адрес и порт сервера findface-facerouter, который получает обнаруженные объекты из findface-video-worker. В FindFace Multi функции findface-facerouter выполняет компонент findface-security. Значение по умолчанию: http://127.0.0.1:18820/v0/frame.
play_speed	Если меньше нуля, то скорость не ограничивается. В остальных случаях поток читается со скоростью play_speed. Не применимо для потоков с камер видеонаблюдения.
disable_drops	Включает/отключает отправку в компонент findface-facerouter всех подходящих объектов без пропусков. По умолчанию, если findface-video-worker не обладает достаточными ресурсами для обработки всех кадров с объектами, он отбрасывает некоторые из них. Если данная опция активна, findface-video-worker помещает лишние кадры в очередь, чтобы обработать их впоследствии. Значение по умолчанию: false.
imotion_threshold	Минимальная интенсивность движения, которая будет регистрироваться детектором движения. Пороговое значение определяется эмпирически. Реперные точки: 0 = детектор выключен, 0.002 = значение по умолчанию, 0.05 = минимальная интенсивность слишком высока, чтобы зарегистрировать движение.
router_timeout	Время ожидания в миллисекундах ответа от компонента findface-facerouter (findface-security в стандартной конфигурации FindFace Multi) на API-запрос компонента findface-video-worker. Если время ожидания истекло, регистрируется ошибка. Значение по умолчанию: 15000.
router_verify_ssl	Включает/отключает проверку подписи SSL-сертификата при взаимодействии по https findface-video-worker с компонентом findface-facerouter (findface-security в стандартной конфигурации FindFace Multi). Значение по умолчанию: true. Если false, может быть принят самоподписанный сертификат.
router_headers	Массив дополнительных заголовков в POST-запросе с изображением объекта в формате ["ключ=значение"]. По умолчанию дополнительные заголовки не передаются.
router_body	Массив дополнительных полей в POST-запросе с изображением объекта в формате ["ключ=значение"]. По умолчанию дополнительные поля не передаются.
ffmpeg_params	Список ffmpeg-параметров видеопотока со значениями в виде массива ключ=значение: ["rtsp_transport=tcp", .., "ss=00:20:00"]. Полный список параметров на сайте FFMPEG . Значение по умолчанию: параметры не указаны.
ffmpeg_format	Передаёт формат FFMPEG (mxg, flv и т. д.), если он не может быть автоматически определен.
use_stream_timestamp	Если true, отправлять на сервер временные метки полученные из потока. Если false, отправлять текущие дату и время.
start_stream_timestamp	Передавать указанное количество секунд к временным меткам из потока.
rot	Включает детектирование и отслеживание объектов только внутри заданной прямоугольной области WxH+X+Y. Используйте данную опцию, чтобы уменьшить нагрузку на findface-video-worker. По умолчанию область не задана.
video_transform	Изменить ориентацию видеокadra сразу после декодирования. Значения (регистронезависимые, в скобках JPEG-Exif Orientation Tag): None (1), FlipHorizontal (2), Rotate180 (3), FlipVertical (4), Transpose (5), Rotate90 (6), Transverse (7), Rotate270 (8). Значение по умолчанию: не задано.
enable_recorder	Включает запись видео для Видеоманитофона (должен быть установлен).

Для каждого типа детектора (лицо, тело, автомобиль) для настройки доступны следующие параметры:

Опция	Описание
<code>filter_min_quality</code>	Минимальное пороговое значение качества изображения объекта. Значение по умолчанию: зависит от типа объекта. Не меняйте значение по умолчанию без предварительной консультации с нашими техническими экспертами (support@ntechlab.com).
<code>filter_min_size</code>	Определяет минимальный размер объекта в пикселях. Вычисляется как корень квадратный из площади bbox объекта. Объекты меньшего размера не отправляются. Значение по умолчанию: 1.
<code>filter_max_size</code>	Определяет максимальный размер объекта в пикселях. Вычисляется как корень квадратный из площади bbox объекта. Объекты большего размера не отправляются. Значение по умолчанию: 8192.
<code>roi</code>	Включает отправку на Сервер объектов, обнаруженных только внутри интересующей области WxH+X+Y. По умолчанию область не задана.
<code>fullframe_crop</code>	Отправлять отправляемые полные кадры по ROT (региону отслеживания). Значение по умолчанию: false.
<code>fullframe_usepng</code>	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. Значение по умолчанию: false (отправлять в формате JPEG).
<code>jpeg_quality</code>	Качество сжатия исходного кадра в JPEG. Значение по умолчанию: 95 % от исходного размера.
<code>overall_only</code>	Буферный режим. Отправлять для объекта один кадр наилучшего качества. Значение по умолчанию: true (CPU), false (GPU).
<code>realtime_post_processing</code>	Обработка изображения объекта сразу после его появления в поле зрения видеокамеры (режим реального времени). Значение по умолчанию: false.
<code>realtime_post_interval</code>	Интервал режима реального времени. Период времени в секундах, в течение которого детектор объектов выбирает лучший кадр из трека и отправляет его в компонент <code>findface-facerouter</code> . Значение по умолчанию: 1.
<code>realtime_post_threshold</code>	Порог качества реального времени. Включает отправку лучшего кадра в течение периода времени <code>realtime_post_interval</code> . Если false, лучший кадр ищется динамически в порядке возрастания качества. Значение по умолчанию: false.
<code>track_interpolation</code>	Использовать пропущенные изображения объектов в треке. Например, если в кадрах №1 and №4 были обнаружены объекты, а в №2 и №3 нет, система воссоздаст отсутствующие изображения объектов №2 и №3 на основании данных №1 и №4. Включение данной опции позволяет добиться увеличения качества детектирования за счет производительности. Значение по умолчанию: true.
<code>track_miss_incomplete</code>	Завершит трек, если в нем не было задетектировано новых изображений объекта в течение заданного времени (секунды). Значение по умолчанию: 1.
<code>track_max_duration</code>	Максимальное количество кадров в треке, по достижении которого трек будет принудительно закрыт. Включите опцию, чтобы автоматически завершать “вечные треки”, например, треки детектирования объектов с рекламных носителей. Значение по умолчанию: 0 (опция отключена).
<code>track_send_history</code>	Отправлять историю трека. Значение по умолчанию: false.
<code>post_best_track</code>	Отправлять полные кадры для обнаруженных объектов. Значение по умолчанию: true.
<code>post_best_track_normalize</code>	Нормализованные изображения для обнаруженных объектов. Значение по умолчанию: true.
<code>post_first_track</code>	Отправлять первый кадр трека. Значение по умолчанию: false.
<code>post_last_track</code>	Отправлять последний кадр трека. Значение по умолчанию: false.

2. Файл конфигурации `findface-video-worker` `/etc/findface-video-worker-cpu.ini` или `/etc/findface-video-worker-gpu.ini`, в зависимости от используемого типа ускорения.

Пользовательская настройка `findface-video-worker` на CPU/GPU выполняется с использованием следующих параметров:

CPU	GPU	Описание
<code>batch_size</code>		Отправлять лица в пакетах указанного размера.
<code>capacity</code>		Максимальное количество видеопотоков, обрабатываемых <code>findface-video-worker</code> .
Н/п	<code>cpu</code>	При необходимости декодировать видео на CPU.
Н/п	<code>device_number</code>	Номер используемого GPU-устройства.
<code>exit_on_first_finished</code>		Если указан <code>input</code>) Выйти после завершения первого job-задания.
<code>input</code>		Обрабатывать видеопотоки из файла, игнорируя данные потоков, поступающие от <code>findface-video-manager</code> .
<code>labels</code>		Метки, используемые для привязки экземпляра <code>findface-video-worker</code> к определенной группе камер. См. <i>Привязка группы камер к экземпляру findface-video-worker</i> .
<code>mgr-cmd</code>		Команда для получения IP-адреса компонента <code>findface-video-manager</code> (опционально вместо <code>mgr-static</code>).
<code>mgr-static</code>		IP-адрес сервера <code>findface-video-manager</code> , который обеспечивает <code>findface-video-worker</code> настройками и списком видеопотоков для обработки.
<code>metrics_port</code>		Порт HTTP-сервера для отправки метрик. Если 0, метрики не отправляются.
<code>min_size</code>		Минимальный обнаруживаемый размер объекта.
<code>ntls-addr</code>		IP-адрес и порт сервера <code>findface-ntls</code> .
<code>resize_scale</code>		Масштабировать видеокадры с заданным коэффициентом.
<code>resolutions</code>		Предварительно инициализируйте <code>findface-video-worker</code> для конкретных разрешений. Например, «640x480;1920x1080».
<code>save_dir</code>		(Для отладки) Сохранять обнаруженные объекты в заданный каталог.
<code>streamer -> port, url</code>		IP-адрес и порт для доступа к <i>видеостене</i> .
<code>use_time_from_sei</code>		(Для MPEG-2) Используйте временные метки SEI (supplemental enhancement information).

При необходимости вы также можете включить модели нейронных сетей и нормализаторы для детекции силуэтов, автомобилей и liveness. Подробные пошаговые инструкции можно найти в следующих разделах:

- *Распознавание живых лиц в реальном времени (Liveness)*
- *Распознавание силуэтов и их атрибутов*
- *Распознавание автомобилей и их атрибутов*

См. также:

Настройка Видеомэгнитофона

Job-задания

Сервис `findface-video-manager` выдает `findface-video-worker` так называемое job-задание, задачу на обработку видео, которая содержит параметры конфигурации и сведения о видеопотоке.

Пример job-задания приведен [здесь](#).

Каждое job-задание имеет следующие параметры:

- `id`: id job-задания.
- `enabled`: статус активности.
- `url`: URL/адрес видеопотока или файла для обработки.
- `labels`: метки, по которым будет осуществляться обработка обнаруженных лиц в компоненте `findface-facerouter` (`findface-security` в стандартной конфигурации FindFace Multi).
- `single_pass`: если `true` (по умолчанию `false`), то не перезапускать обработку потока в случае ошибки.
- `stream_settings`: параметры видеопотока, дублирующие *параметры* в файле конфигурации `/etc/findface-video-manager.conf` (обладая при этом большим приоритетом).
- `status`: статус job-задания.
- `msg`: дополнительная информация о статусе job-задания.
- `statistic`: статистика выполнения job-задания (продолжительность процесса обработки, количество отправленных и неотправленных объектов, кадровая частота обработки, количество обработанных и пропущенных кадров, время начала обработки и т. д.).
- служебная и справочная информация о job-задании.

findface-ntls

Локальный сервер лицензирования `findface-ntls` – это сервис, который устанавливается на выбранном физическом сервере и служит для верификации лицензии FindFace. Для верификации используются следующие источники:

- Глобальный сервер лицензий NtechLab. Служит для лицензирования в сетях с доступом в интернет, в том числе с доступом через прокси-сервер.
- Ключ аппаратной защиты. Служит для лицензирования в закрытых сетях.

Используйте веб-интерфейс FindFace Multi (*Настройки -> Лицензия*) для управления `findface-ntls` следующим образом:

- просмотр списка приобретенных функций,
- просмотр списка текущих ограничений,
- загрузка файла лицензии,
- просмотр списка подключенных в данный момент компонентов.

Лицензируются следующие компоненты:

- `findface-tarantool-server`,
- `findface-extraction-api`,
- `findface-video-manager`,
- `findface-video-worker`.

Важно: После разрыва соединения между сервером лицензирования `findface-ntls` и лицензируемым компонентом или между сервером лицензирования `findface-ntls` и глобальным сервером лицензирования, время автономной работы компонентов составляет 4 часа. Обратитесь к своему менеджеру, чтобы продлить период автономной работы (до 2-х дней).

Настройка компонента `findface-ntls` выполняется через файл конфигурации `/etc/findface-ntls.cfg`. Вы можете просмотреть его исходное содержимое [здесь](#).

Пользовательская настройка `findface-ntls` выполняется с использованием следующих параметров:

Параметр	Описание
<code>license_dir</code>	Путь к каталогу для хранения файла лицензии.
<code>listen</code>	IP-адрес сервера, с которого осуществляется обращение лицензируемых компонентов в <code>findface-ntls</code> . Для того чтобы разрешить обращение с любого IP-адреса, установите значение <code>0.0.0.0:3133</code> .
<code>proxy</code>	IP-адрес и порт прокси-сервера (опционально).
<code>ui</code>	IP-адрес сервера, с которого будет доступен веб-интерфейс <code>findface-ntls</code> . Для того чтобы разрешить доступ со всех адресов, измените значение на <code>“0.0.0.0”</code> .

`findface-security`

Компонент `findface-security` обеспечивает доступ конечного пользователя к функциям ядра FindFace. Отвечает за взаимодействие между ядром FindFace Core и веб-интерфейсом, а также функционирование системы как единого целого.

Компонент `findface-security` также выполняет функции компонента `findface-facerouter` (часть ядра FindFace), задавая правила обработки обнаруженных объектов. Он получает рамку с объектом и его нормализованное изображение вместе с исходным кадром и другими данными (например, датой и временем детекции) от сервиса `findface-video-worker` и перенаправляет их для дальнейшей обработки в сервис `findface-sf-api`.

Настройка компонента `findface-security` выполняется через файл конфигурации `/etc/findface-security/config.py`. Вы можете просмотреть его исходное содержимое [здесь](#).

Файл `/etc/findface-security/config.py` содержит подробные комментарии по каждой настройке. Руководствуйтесь ими при конфигурировании `findface-security`.

`findface-facerouter` и пользовательские плагины

Важно: Компонент `findface-facerouter` не входит в состав стандартной конфигурации FindFace Multi. При необходимости используйте его для интеграции.

Компонент `findface-facerouter` представляет собой сервис, через который задаются правила обработки обнаруженных на видео объектов. Правила задаются в виде пользовательских плагинов.

Компонент `findface-facerouter` принимает нормализованные изображения объектов вместе с исходным кадром и другой информацией (например, датой и временем детекции) от компонента `findface-video-worker`.

В общем случае позволяет обрабатывать полученные объекты произвольным способом, в том числе отправлять их напрямую в партнерское приложение. В FindFace Multi функции `findface-facerouter` выполняет `findface-security`, который перенаправляет объекты в компонент `findface-sf-api`.

Полный список функций:

- задание правил обработки обнаруженных на видео объектов,
- перенаправление обнаруженных объектов в компонент `findface-sf-api` или другой сервис (в том числе стороннее приложение) для последующей обработки.

Настройка компонента `findface-facerouter` выполняется через файл конфигурации `/etc/findface-facerouter.py`. Вы можете просмотреть его исходное содержимое [здесь](#).

Пользовательская настройка `findface-facerouter` выполняется с использованием следующих параметров:

Параметр	Описание
<code>sfapi_url</code>	IP-адрес и порт сервера <code>findface-sf-api</code> .
<code>plugin_dir</code>	Каталог с плагинами, определяющими правила обработки обнаруженных на видео объектов.

Развертывание `findface-facerouter` в FindFace Multi

Для развертывания компонента `findface-facerouter` выполните следующие действия:

1. Установите `findface-facerouter` либо из *консольного инсталлятора*, либо из apt-репозитория следующим образом:

```
sudo apt update
sudo apt install -y findface-facerouter
```

2. Откройте файл конфигурации `/etc/findface-facerouter.py`.

```
sudo vi /etc/findface-facerouter.py
```

3. Если компоненты `findface-facerouter` и `findface-sf-api` установлены на разных физических серверах, раскомментируйте параметр `sfapi_url` и укажите в нем IP-адрес сервера `findface-sf-api`.

```
sfapi_url = 'http://localhost:18411'
```

4. Откройте файл конфигурации `/etc/findface-security/config.py`. В параметре `ROUTER_URL` актуализируйте IP-адрес и порт `findface-facerouter` (по умолчанию порт 18820). IP-адрес указывается внешний или внутренний в зависимости от сети, по которой `findface-video-worker` взаимодействует с `findface-facerouter`.

```
sudo vi /etc/findface-security/config.py

...
FFSECURITY = {
    'ROUTER_URL': 'http://127.0.0.1:18820/v0/frame?',
```

5. Добавьте сервис `findface-facerouter` в автозагрузку Ubuntu и запустите сервис.

```
sudo systemctl enable findface-facerouter.service && sudo systemctl start findface-  
facerouter.service
```

6. Перезапустите сервис `findface-security`.

```
sudo systemctl restart findface-security.service
```

Настройка `findface-facerouter` на использование плагинов

Важно: Обязательно предварительно *измените* структуру биометрической базы Tarantool в соответствии с правилами обработки, заданными в плагинах.

Важно: Компонент `findface-facerouter` должен быть *установлен и настроен*.

Для настройки компонента `findface-facerouter` на использование плагинов выполните следующие действия:

1. Поместите плагин в каталог по вашему выбору. Все используемые плагины должны находиться в одном каталоге.
2. Откройте файл конфигурации `/etc/findface-facerouter.py`. Раскомментируйте параметр `plugin_dir` и укажите каталог с плагином.

Предупреждение: Содержимое `findface-facerouter.py` должно представлять собой синтаксически корректный код Python.

```
sudo vi /etc/findface-facerouter.py  
  
plugin_dir = '/etc/findface/plugins/'
```

3. Перезапустите `findface-facerouter`.

```
sudo systemctl restart findface-facerouter.service
```

Принципы написания плагина

В этом разделе:

- *Архитектура плагина*
- *Метод `preprocess`*
- *Метод `process`*
- *Метод `shutdown`*

Архитектура плагина

После того как компонент `findface-video-worker` обнаруживает лицо, он отправляет его в компонент `findface-facerouter` в виде HTTP API запроса. Для обработки запроса каждый плагин должен экспортировать функцию `activate(app, ctx, plugin_name, plugin_source)`.

Параметры функции `activate`:

- `app`: сущность `tornado.web.Application` компонента `findface-facerouter`.
- `ctx`: контекст, передаваемый плагину при активации.
- `plugin_name`: имя активируемого плагина.
- `plugin_source`: объект источника, из которого загружается плагин.

При активации плагину передается следующий контекст:

1. `request.ctx.sfapi`: настроенный экземпляр `ntech.sfapi_client.Client`, к которому можно обращаться напрямую для обработки результата видеодетекции (создание новой галереи, добавление лица в галерею и т. д.).
2. `plugins`: `OrderedDict` со всеми плагинами (`key`: имя плагина, `value`: результат, возвращенный функцией `activate`).
3. `idgen`: генератор id, который может вызываться как `ctx.idgen()`.

Функция `activate(app, ctx, plugin_name, plugin_source)` должна вернуть объект со следующими методами:

1. `preprocess`,
2. `process`,
3. `shutdown` (опционально).

Метод `preprocess`

В данном методе плагин решает, интересует ли его полученное лицо, и если да, возвращает кортеж или список, содержащий одну или несколько строк: `'facen'`, `'gender'`, `'age'`, `'emotions'`, что соответственно означает, что нужно извлечь вектор признаков, распознать пол, возраст и/или эмоции. Если возвращенные кортеж или список непусты, компонент `findface-facerouter` перенаправляет обнаруженное лицо в компонент `findface-sf-api` в запросе `/detect POST` с соответствующими параметрами в query string (`facen=on, gender=on, age=on, emotions=on`).

Синтаксис базового метода `preprocess`, от которого следует наследоваться (см. класс `Plugin`):

```
preprocess(self, request: FrHTTPRequest, labels: Mapping[str, str]) → Tuple[str]
```

Параметры

- `FrHTTPRequest` (`tornado.httpserver.HTTPRequest`) – HTTP API запрос, который включает в себя дополнительный аргумент `params`
- `labels` (`dictionary`) – пользовательский набор меток кадра, который задается в параметрах задания для компонента `findface-video-worker` и затем присваивается кадру

Аргумент `params` `FrHTTPRequest` содержит следующие поля:

Параметры

- `photo` (`bytes`) – кадр с лицом в формате JPEG

- `face0 (bytes)` – нормализованное изображение лица
- `bbox (list of integers [[x1,y1,x2,y2]]`, where `x1`: x coordinate of the top-left corner, `y1`: y coordinate of the top-left corner, `x2`: x coordinate of the bottom-right corner, `y2`: y coordinate of the bottom-right corner) – координаты рамки с лицом в кадре
- `cam_id (string)` – id видеокамеры
- `timestamp (datetime.datetime)` – временная метка кадра
- `detectorParams (dictionary)` – словарь со служебно-отладочной информацией от детектора
- `bs_type (string)` – режим поиска лучшего кадра. Доступные опции: `overall` (буферный режим: на сервер отправляется кадр, который был лучшим за весь период нахождения лица в поле зрения видеокамеры), `realtime` (режим реального времени: на сервер отправляются кадры, считающиеся лучшими в последовательных интервалах).
- `labels (dictionary)` – (дублирует `params.labels`) пользовательский набор меток кадра, который задается в параметрах задания для компонента `findface-video-worker` и затем присваивается кадру

Решение об обработке лица принимается на основании данных из `request.params`, в том числе пользовательского набора меток, а также из любых других соображений.

Метод `process`

Данный метод вызывается, если метод `preprocess` вернул непустой кортеж или список (`'facen'`, `'gender'`, `'age'`, `'emotions'`). После того, как компонент `findface-sf-api` вернул ответ с результатом детекции (см. запрос `/detect POST`) со всеми запрошенными параметрами лица, компонент `findface-facerouter` вызывает метод `process` плагина для выполнения собственно обработки лица.

Для выполнения обработки лица плагин использует `request.ctx.sfapi`.

Синтаксис базового метода `process`, от которого следует наследоваться (см. класс `Plugin`):

```
process(self, request: FrHTTPRequest, photo: bytes, bbox: List[int], event_id: int, detection: DetectFace)
```

Метод `shutdown`

Данный метод вызывается только перед завершением работы компонента `findface-facerouter`.

Синтаксис базового метода `shutdown`, от которого следует наследоваться (см. класс `Plugin`):

```
shutdown(self)
```

Классы и методы плагинов

В этом разделе:

- Базовые классы
- Классы объектов
- Обнаружение лица и работа с галереями
- Фильтры для поиска по базе данных
- Отображение сообщений об ошибках

Базовые классы

```
class facerouter.plugin.Plugin
```

Данный класс предоставляет базовые методы для написания плагина, описанные в разделе *Принципы написания плагина*. Пользовательский класс, выполняющий роль оболочки для плагина, должен наследовать от класса `Plugin`.

```
preprocess(self, request: FrHTTPRequest, labels: Mapping[str, str]) → Tuple[str]
```

Возвращает кортеж, включающий в себя одну или несколько строк: 'facen', 'gender', 'age', 'emotions', что соответственно означает, что компонент `findface-facerouter` должен запросить у компонента `findface-sf-api` извлечение вектора признаков, распознать пол, возраст и/или эмоции.

Параметры

- `FrHTTPRequest` (*tornado.httpserver.HTTPRequest*) – HTTP API запрос, который включает в себя дополнительный аргумент `params`
- `labels` (*dictionary*) – пользовательские метки из `request.params`

Результат

одна или несколько строк 'facen', 'gender', 'age', 'emotions'

Тип результата

tuple

Аргумент `params` `FrHTTPRequest` содержит следующие поля:

Параметры

- `photo` (*bytes*) – кадр с лицом в формате JPEG
- `face0` (*bytes*) – нормализованное изображение лица
- `bbox` (list of integers `[[x1,y1,x2,y2]]`, where `x1`: x coordinate of the top-left corner, `y1`: y coordinate of the top-left corner, `x2`: x coordinate of the bottom-right corner, `y2`: y coordinate of the bottom-right corner) – координаты рамки с лицом в кадре
- `cam_id` (*string*) – id камеры
- `timestamp` (*datetime.datetime*) – временная метка кадра
- `detectorParams` (*dictionary*) – словарь со служебно-отладочной информацией от детектора

- `bs_type` (*string*) – режим поиска лучшего кадра. Доступные опции: `overall` (буферный режим: на сервер отправляется кадр, который был лучшим за весь период нахождения лица в поле зрения видеокамеры), `realtime` (режим реального времени: на сервер отправляются кадры, считающиеся лучшими в последовательных интервалах).
- `labels` (*dictionary*) – (дублирует `params.labels`) пользовательский набор меток кадра, который задается в параметрах задания для компонента `findface-video-worker` и затем присваивается кадру

`process(self, request: FrHTTPRequest, photo: bytes, bbox: List[int], event_id: int, detection: DetectFace)`

Принимает атрибуты обнаруженного лица.

Параметры

- `request` (*tornado.httpserver.HTTPRequest*) – HTTP API-запрос от `findface-video-worker`
- `photo` (*bytes*) – кадр с лицом в формате JPEG из `request.params`
- `bbox` (list of integers `[[x1,y1,x2,y2]]`, where `x1`: x coordinate of the top-left corner, `y1`: y coordinate of the top-left corner, `x2`: x coordinate of the bottom-right corner, `y2`: y coordinate of the bottom-right corner) – координаты рамки с лицом в кадре из `request.params`
- `event_id` (*uint64*) – id обнаруженного на видео лица (автоматически задается компонентом `findface-facerouter` при получении лица от `findface-video-worker`). Может использоваться в качестве пользовательского идентификатора лица в базе данных.
- `detection` (*objects.DetectFace*) – результат детекции, полученный от компонента `findface-sf-api`, включающий в себя запрошенные параметры лица, такие как вектор признаков, пол, возраст, эмоции.

Результат

н/п

Тип результата

н/а

`shutdown(self)`

Данный метод вызывается только перед завершением работы компонента `findface-facerouter`.

Параметры

н/п

Результат

н/п

Классы объектов

`class objects.BBox`

Представляет собой координаты рамки с лицом.

`class objects.DetectFace`

Представляет собой результат детекции со следующими полями:

Параметры

- `id (string)` – id результата детекции в memcached
- `bbox (objects.Bbox)` – координаты рамки с лицом
- `features (dictionary)` – информация о поле (gender), возрасте (age) и эмоциях (emotions) (опционально)

`class objects.DetectResponse`

Представляет собой список объектов `objects.DetectionFace` с дополнительным полем `orientation`, содержащим информацию об ориентации EXIF лица.

Параметры

`orientation (EXIF orientation)` – ориентация обнаруженного лица

`class objects.FaceId(namedtuple('FaceId', ('gallery', 'face')))`

Представляет собой объект пользовательского идентификатора лица в галерее.

Параметры

- `gallery (string)` – имя галереи
- `face (integer)` – пользовательский идентификатор лица в галерее

`class objects.Face`

Представляет собой результат поиска лица в базе данных по биометрическому образцу

Параметры

- `id (objects.FaceId)` – объект Faceid
- `features (dictionary)` – информация о поле, возрасте и эмоциях
- `meta (dictionary)` – метаданные лица
- `confidence (float)` – степень схожести лица с заданным биометрическим образцом

`class objects.ListResponse`

Представляет собой список объектов `objects.Face` (т. е. список результатов поиска по биометрическому образцу) с дополнительным полем `next_page`, содержащим информацию о следующей странице с результатами.

Параметры

`next_page (string)` – курсор следующей страницы с результатами поиска

Обнаружение лица и работа с галереями

`class ntech.sfapi_client.client.Client`

Предоставляет базовые методы для обнаружения лиц на изображении и работы с галереями.

`detect(self, *, url=None, image=None, facen=False, gender=False, age=False, emotions=False, return_facen=False, autorotate=False, detector: str = None, timeout=None) → DetectResponse`

Обнаруживает лица на изображении и возвращает обнаруженные лица.

Параметры

- `url (URL)` – URL изображения, если вы передаете общедоступное изображение из интернета
- `image (bytes)` – файл PNG/JPG/WEBP, если вы передаете изображение в виде файла
- `facen (boolean)` – извлечь вектор признаков из обнаруженного лица. Для сохранения результата детекции в `memcached` передайте `facen=True`.
- `gender (boolean)` – извлечь и вернуть информацию о поле
- `age (boolean)` – извлечь и вернуть информацию о возрасте
- `emotions (boolean)` – извлечь и вернуть информацию об эмоциях
- `return_facen (boolean)` – вернуть вектор признаков в результате работы метода
- `autorotate (boolean)` – автоматически повернуть изображение в 4-х разных ориентациях для обнаружения лиц в каждой их них. Пересекающиеся направления с `IOU > 0.5` будут объединены
- `detector (boolean)` – может принимать значение `nnd` или `normalized`. Детектор `normalized` используется для обработки нормализованных изображений, например, поступающих от видеодетектора лиц
- `timeout (number)` – максимальное время ожидания ответа от ядра FindFace в секундах (если `none`, используется время ожидания ответа, заданное по умолчанию)

Результат

Результат детекции

Тип результата

Объект `DetectorResponse`

`gallery(self, name)`

Возвращает объект `sfapi_client.Gallery` для последующей с ним работы (например, получения списка лиц).

Параметры

`name (string)` – имя галереи

Результат

объект типа «галерея»

Тип результата

`sfapi_client.Gallery`


```
list_galleries(self, timeout=None):
```

Возвращает список галерей.

Параметры

`timeout` (*number*) – максимальное время ожидания ответа от ядра FindFace в секундах (если `none`, используется время ожидания ответа, заданное по умолчанию)

Результат

список галерей со свойствами `name` (имя галереи, строка) и `number` (количество лиц в галереи, число)

Тип результата

`list of GalleryListItem`

```
class ntech.sfapi_client.gallery.Gallery
```

Предоставляет методы для работы с галереями и лицами в них.

```
list(self, *, filters: Iterable[filters.Filter] = None, limit: int = 1000, sort: str = '', page=None,
      ignore_errors=False, timeout=None) → ListResponse
```

Возвращает объект типа список с лицами из галереи, удовлетворяющими заданным фильтрам. Возвращаемый объект типа список содержит дополнительное свойство `next_page`, которое может использоваться как значение параметра `page` в последующих запросах.

Параметры

- `filters` (`sfapi_client.filters.Filter`) – список фильтров
- `limit` (*integer*) – максимальное количество лиц в ответе
- `sort` (*string*) – метод сортировки лиц. Возможные значения: `id` (по возрастанию `id`), `-id` (по убыванию `id`), `-confidence` (по убыванию степени схожести лиц). Сортировка по `id` возможна только при отключенном фильтре `facen`, который задает вектор признаков для поиска в базе данных (т.н. идентификация лица). Наоборот, сортировка по степени схожести лиц (`confidence`) возможна только при включенном фильтре `facen`. По умолчанию метод использует сортировку по возрастанию `id` (вектор признаков не задан) и по убыванию степени схожести лиц (вектор признаков задан).
- `page` – вернуть результаты, начиная с указанной страницы. Номер следующей страницы с результатами возвращается в ответе сервера в виде `next_page`.
- `ignore_errors` (*boolean*) – Игнорировать ошибку обращения к базе данных, если поиск по галерее выполняется, когда некоторые сервера базы данных недоступны. В этом случае поиск будет выполнен с использованием доступных серверов.
- `timeout` (*number*) – максимальное время ожидания ответа от ядра FindFace в секундах (если `none`, используется время ожидания ответа, заданное по умолчанию)

Результат

список с лицами из галереи, удовлетворяющими заданным фильтрам.

Тип результата

`ListResponse object`

```
add(self, new_id: Union[int, Callable], source: Union[DetectFace, Face, str], *, meta: Dict[str,
Union[int, str, List[str]]] = None, regenerate_attempts=None, timeout=None) → Face
```

Создает лицо в галерее.

Параметры

- `new_id` (*integer or callable*) – пользовательский идентификатор лица в базе данных. Может быть (`async`) `callable`, который возвращает `id`. Для генерации `id` может использоваться функция `ctx.idgen()` из контекста.
- `source` (*`sfapi_client.DetectFace`, `sfapi_client.Face`, `sfapi_client.FaceId`, or `string`*) – источник, из которого лицо добавляется в базу данных, может представлять собой лицо в базе данных или результат детекции.
- `meta` (*dictionary*) – метаданные лица. Ключи могут быть строками, а значения – целыми числами, строками или списками строк. Перед добавлением метаданных в базу данных должна быть создана соответствующая структура.
- `regenerate_attempts` – количество попыток генерации уникального `id` функцией `ctx.idgen()`, если `new_id` `callable`.
- `timeout` (*number*) – максимальное время ожидания ответа от ядра FindFace в секундах (если `none`, используется время ожидания ответа, заданное по умолчанию)

Результат

представление созданного лица

Тип результата

Face object

`delete(self, face: Union[Face, int], timeout=None) → None`

Удаляет лицо из галереи.

Параметры

- `face` (*`sfapi_client.Face`, `sfapi_client.FaceId` or `id in integer`*) – лицо, которое нужно удалить из базы данных
- `timeout` (*number*) – максимальное время ожидания ответа от ядра FindFace в секундах (если `none`, используется время ожидания ответа, заданное по умолчанию)

Результат

None

`get(self, face: Union[Face, int], timeout=None) → Face`

Возвращает лицо из галереи.

Параметры

- `face` (*`sfapi_client.Face`, `sfapi_client.FaceId` or `id in integer`*) – лицо, которое нужно извлечь из базы данных
- `timeout` (*number*) – максимальное время ожидания ответа от ядра FindFace в секундах (если `none`, используется время ожидания ответа, заданное по умолчанию)

Результат

представление лица

Тип результата

Face object

`create(self, timeout=None) → None`

Создает галерею в `findface-sf-api` в виде объекта `sfapi_client.Gallery`. Объект `sfapi_client.Gallery` представляет собой промежуточный объект, и для работы с ним не требуется наличия галереи на сервере.

Параметры

`timeout` (*number*) – максимальное время ожидания ответа от ядра FindFace в секундах (если `none`, используется время ожидания ответа, заданное по умолчанию)

Результат

None

`drop(self, timeout=None) → None:`

Удаляет галерею из `findface-sf-api`.

Параметры

`timeout` (*number*) – максимальное время ожидания ответа от ядра FindFace в секундах (если `none`, используется время ожидания ответа, заданное по умолчанию)

Результат

None

`update(self, face: Union[Face, str], *, meta: Dict[str, Union[int, str, List[str]]] = None, timeout=None) → Face`

Редактирует метаданные лица в галерее.

Параметры

- `face` (`sfapi_client.Face`, `sfapi_client.FaceId` or `id in integer`) – лицо, метаданные которого нужно заменить в базе данных
- `meta` (*dictionary*) – метаданные лица, которые нужно заменить. Ключи могут быть строками, а значения – целыми числами, строками или списками строк. Если поля `meta` не передаются или `null`, они не изменяются в базе данных.
- `timeout` (*number*) – максимальное время ожидания ответа от ядра FindFace в секундах (если `none`, используется время ожидания ответа, заданное по умолчанию)

Результат

представление измененного лица

Тип результата

Face object

Фильтры для поиска по базе данных

`class ntech.sfapi_client.filters.Filter`

Общий класс. Представляет собой сводный список фильтров (со значениями), которые должны быть применены к содержимому галереи.

`serialize(self)`

Метод для передачи списка фильтров со значениями в компонент `findface-sf-api`.

Результат

имена и значения заданных фильтров

Тип результата

`tuple („filtername“, [«value1», «value2»])`.

`class ntech.sfapi_client.filters.Id`

Предоставляет методы для фильтрации содержимого галереи по id. Для использования фильтра нужно напрямую вызвать соответствующий classmethod, не создавая экземпляр класса.

`classmethod lte(cls, value: int) → Filter`

Фильтр LTE. Выбрать все лица с id, меньшим или равным указанному.

Параметры

`value (integer)` – значение id

Результат

имя фильтра (LTE) и его значение.

Тип результата

объект класса Filter.

Пример: `Id.lte(1234)` выбирает лица с id, меньшим или равным 1234.

`classmethod gte(cls, value: int) → Filter`

Фильтр GTE. Выбрать все лица с id, большим или равным указанному.

Параметры

`value (integer)` – значение id

Результат

имя фильтра (GTE) и его значение.

Тип результата

объект класса Filter.

Пример: `Id.gte(1234)` выбирает лица с id, большим или равным 1234.

`classmethod oneof(cls, *value: Union[int]) → Filter`

Фильтр IN. Выбрать лица с id из заданной последовательности.

Параметры

`value (list of integers)` – список значений id

Результат

имя фильтра (IN) и его значение.

Тип результата

объект класса Filter.

Пример: `Id.oneof(1234, 5678)` выбирает лицо с id 1234 и/или 5678.

`class ntech.sfapi_client.filters.Meta`

Предоставляет методы для фильтрации содержимого галереи по метаданным. Для использования фильтра нужно напрямую вызвать соответствующий метод, не создавая экземпляр класса.

`classmethod lte(self, value: Union[str, int]) → Filter`

Фильтр LTE. Выбрать все лица, у которых определенная строка в метаданных меньше или равна указанному значению.

Параметры

`value (string or integer)` – значение строки с метаданными

Результат

имя фильтра (LTE) и его значение.

Тип результата

объект класса `Filter`.

Пример: `Meta('foo').lte(1234)` выбирает лица с мета-строкой `foo`, имеющей значение меньше или равное 1234.

`classmethod gte(self, value: Union[str, int]) → Filter`

Фильтр GTE. Выбрать все лица, у которых определенная строка в метаданных больше или равна указанному значению.

Параметры

`value` (*string or integer*) – значение строки с метаданными

Результат

имя фильтра (GTE) и его значение.

Тип результата

объект класса `Filter`.

Пример: `Meta('foo').gte(1234)` выбирает лица с мета-строкой `foo`, имеющей значение больше или равное 1234.

`classmethod oneof(self, *value: Union[str, int]) → Filter`

Фильтр IN. Выбрать все лица, у которых определенная строка в метаданных совпадает с одним из значений из заданной последовательности.

Параметры

`value` (*list of strings or integers*) – список строк с метаданными

Результат

имя фильтра (IN) и его значение.

Тип результата

объект класса `Filter`.

Пример: `Meta.oneof(1234, 5678)` выбирает лица с мета-строкой, имеющей значение 1234 и/или 5678.

`classmethod subset(self, *value: str) → Filter`

Фильтр SUBSET. Выбрать все лица, у которых определенная строка содержит все значения из указанной последовательности.

Параметры

`value` (*list of strings or integers*) – список строк с метаданными

Результат

имя фильтра (SUBSET) и его значение.

Тип результата

объект класса `Filter`.

Пример: `Meta('foo').subset("male", "angry")` выбирает лица с мета-строкой `foo`, содержащей все значения из последовательности ["male", "angry"].

`class ntech.sfapi_client.filters.Detection(Filter)`

Предоставляет метод для идентификации (поиска похожих лиц в базе данных) обнаруженного лица.

`__init__(self, id: Union[str, objects.DetectFace], threshold: float)`

Параметры

- `id` (`objects.DetectFace` or temporary face id in memcached returned by `sfapi_client.Client.detect()`, string) – лицо (результат детекции), которое нужно найти в базе данных
- `threshold` (*float*) – минимальная степень схожести лиц от 0 до 1

Пример: `Detection(det1, 0.77)` выбирает лица, похожие на результат детекции `det1` со степенью схожести, большей или равной 0.77.

```
class ntech.sfapi_client.filters.Face(Filter)
```

Предоставляет метод для поиска лиц в базе данных, похожих на лицо из галереи.

```
__init__(self, id: Union[str, objects.Face], threshold: float)
```

Параметры

- `id` (`objects.Face`, `objects.FaceId` or custom face id in the gallery, string) – лицо из базы данных, которое нужно найти
- `threshold` (*float*) – минимальная степень схожести лиц от 0 до 1

Пример: `Detection(FaceId("gal1", 1234), 0.77)` выбирает лица, похожие на лицо с пользовательским идентификатором `face 1234` из галереи `gal1` со степенью схожести, большей или равной 0.77.

Пример использования нескольких фильтров

```
filters=[filters.Id.gte(123456), filters.Meta('age').gte(45), filters.Meta('camera').  
↳oneof('abc', 'def')]
```

Отображение сообщений об ошибках

```
class sfapi_client.SFApiRemoteError
```

Данное сообщение об ошибке появляется, если ошибка произошла по причине, отличной от сетевого сбоя.

Сообщение об ошибке содержит как минимум два поля:

- `code` — это код ошибки в виде `CAPS_AND_UNDERSCORES`, который может быть использован для автоматического преобразования.
- `reason` — это описание ошибки, предназначенное для прочтения человеком.

Полный список ошибок

Код ошибки	Описание
UNKNOWN_ERROR	Ошибка неизвестного происхождения.
BAD_PARAM	Запрос может быть прочитан, однако некоторые параметры метода недействительны. Данный тип ответа содержит дополнительные атрибуты param и value для описания ошибочных параметров.
CONFLICT	Конфликт.
EXTRACTION_ERROR	Ошибка при извлечении из лица вектора признаков.
LICENSE_ERROR	Конфигурация системы не соответствует лицензии.
MALFORMED_REQUEST	Запрос неправильно сформирован и не может быть прочитан.
OVER_CAPACITY	Превышен размер очередей в компоненте findface-extraction-api .
SOURCE_NOT_FOUND	В параметре from задано несуществующее лицо.
SOURCE_GALLERY_NOT_FOUND	В параметре from задана несуществующая галерея.
STORAGE_ERROR	Биометрическая база данных недоступна.
CACHE_ERROR	Хранилище memcached недоступно.
NOT_FOUND	Подходящие лица не найдены.
NOT_IMPLEMENTED	Функционал не реализован.
GALLERY_NOT_FOUND	Подходящие галереи не найдены.

```
class sfapi_client.SFApiMalformedResponseError
```

Это сообщение об ошибке появляется, если ошибка произошла из-за сбоя в сети, или если Клиент не смог прочитать API-ответ от **findface-sf-api**.

Видеомагнитофон: **findface-video-storage** и **findface-video-streamer-cpu**

Видеомагнитофон является дополнительно устанавливаемой частью ядра FindFace. Видеомагнитофон работает следующим образом:

- Сервис **findface-video-storage** реализует 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-security** выдает информацию об имеющихся видеофрагментах в форме ссылок на Websocket-потoki. Данные ссылки далее используются сервисом **findface-video-streamer**, чтобы передать видео пользователю для просмотра и скачивания.
- После получения запроса от **findface-security-ui** сервис **findface-video-streamer** 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.

Настройка компонента **findface-video-storage** выполняется через файл конфигурации **/etc/findface-video-storage.conf**. Вы можете просмотреть его исходное содержимое [здесь](#).

Настройка компонента **findface-video-streamer-cpu** выполняется через файл конфигурации **/etc/findface-video-streamer-cpu.ini**. Вы можете просмотреть его исходное содержимое [здесь](#).

Для того чтобы настроить Видеомагнитофон, следуйте [инструкции](#).

2.9.2 Файл с параметрами установки

При установке FindFace Multi из инсталлятора параметры установки автоматически сохраняются в файл `/tmp/<findface-installer-*>.json`. Вы можете отредактировать данный файл и использовать его при установке FindFace Multi на других серверах, не отвечая на вопросы инсталлятора повторно.

Совет: Подробная информация об инсталляторе приведена в разделе *Развертывание из консольного инсталлятора*.

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

Пример файла с параметрами установки приведен [здесь](#).

2.9.3 Модели нейронных сетей

В этом разделе вы найдете сводную информацию по моделям нейронных сетей, созданным в нашей лаборатории и используемым в FindFace Multi.

Вы можете найти установленные модели в каталоге `/usr/share/findface-data/models/`.

Важно: При чистой установке биометрической моделью по умолчанию является `kiwi_320`.

Обнаружение лиц

```
ls /usr/share/findface-data/models/facedet/

cheetah.cpu.fnk  cheetah_fast.cpu.fnk  cheetah_fast.gpu.fnk  cheetah.gpu.fnk
```

Нормализация изображения лиц и силуэтов

```
ls /usr/share/findface-data/models/facenorm/

bee_fast.cpu.fnk  bee.v2.gpu.fnk  crop2x.v2_maxsize400.cpu.fnk  crop2x.v2_
↳no_maxsize.gpu.fnk
bee_fast.gpu.fnk  crop1x.v2_maxsize400.cpu.fnk  crop2x.v2_maxsize400.gpu.fnk  cropbbox.
↳v2.cpu.fnk
bee.v2.cpu.fnk  crop1x.v2_maxsize400.gpu.fnk  crop2x.v2_no_maxsize.cpu.fnk  cropbbox.
↳v2.gpu.fnk
```


Распознавание лиц

```
ls /usr/share/findface-data/models/face/

kiwi_160.cpu.fnk  kiwi_320.cpu.fnk
kiwi_160.gpu.fnk  kiwi_320.gpu.fnk
```

Распознавание атрибутов лиц

```
ls /usr/share/findface-data/models/faceattr/

age.v2.cpu.fnk      emotions.v1.cpu.fnk  glasses3.v0.cpu.fnk    liveness.pacs.v0.cpu.
↳fnk  quality.v1.cpu.fnk
age.v2.gpu.fnk      emotions.v1.gpu.fnk  glasses3.v0.gpu.fnk    liveness.pacs.v0.gpu.
↳fnk  quality.v1.gpu.fnk
beard.v0.cpu.fnk    gender.v2.cpu.fnk    liveness.colombo.cpu.fnk  medmask3.v2.cpu.fnk
beard.v0.gpu.fnk    gender.v2.gpu.fnk    liveness.colombo.gpu.fnk  medmask3.v2.gpu.fnk
```

Обнаружение автомобилей

```
ls /usr/share/findface-data/models/cadet/

efreitor.cpu.fnk  efreitor.gpu.fnk
```

Нормализация изображений автомобилей

```
ls /usr/share/findface-data/models/carnorm/

anaferon.v3.cpu.fnk  anaferon.v3.gpu.fnk
```

Распознавание автомобилей

```
ls /usr/share/findface-data/models/carrec/

alonso.cpu.fnk  alonso.gpu.fnk
```

Распознавание атрибутов автомобилей

```
ls /usr/share/findface-data/models/carattr/

carattr.license_plate_quality.v0.cpu.fnk  carattr.quality.v0.cpu.fnk      description.
↳v0.cpu.fnk
carattr.license_plate_quality.v0.gpu.fnk  carattr.quality.v0.gpu.fnk      description.
↳v0.gpu.fnk
carattr.license_plate.v4.cpu.fnk          carattr.special_types.v0.cpu.fnk
carattr.license_plate.v4.gpu.fnk          carattr.special_types.v0.gpu.fnk
```

Обнаружение силуэтов

```
ls /usr/share/findface-data/models/pedet/  
  
glenn_005.cpu.fnk  glenn_005.gpu.fnk  glenny_005_fast.cpu.fnk  glenny_005_fast.gpu.fnk
```

Распознавание силуэтов

```
ls /usr/share/findface-data/models/pedrec/  
  
andariel.cpu.fnk  andariel.gpu.fnk
```

Распознавание атрибутов силуэтов

```
ls /usr/share/findface-data/models/pedattr/  
  
pedattr.clothes_type.v0.cpu.fnk  pedattr.color.v1.cpu.fnk  pedattr.quality.v0.cpu.fnk  
pedattr.clothes_type.v0.gpu.fnk  pedattr.color.v1.gpu.fnk  pedattr.quality.v0.gpu.fnk
```

2.9.4 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 и события.
- Основная база данных системы на основе PostgreSQL, в которой хранятся внутренние данные системы, карточки, учетные записи пользователей и настройки камер.
- Каталог `/var/lib/findface-security/uploads`, в котором хранятся загруженные в карточки фотографии, видеофайлы, полные кадры событий и счетчиков, а также миниатюры объектов.
- Каталог `/var/lib/ffupload/`, в котором хранятся такие артефакты событий, как нормализованные изображения объектов.

Примечание: В данном каталоге также хранятся видеофрагменты, если в системе активирована функция *Видеомагнитофон*.

- (Только вместе с Видеомэгнитофоном) База данных на основе MongoDB, в которой хранится метаданные о видеофрагментах, включая их точное местоположение в хранилище `/var/lib/ffupload/`.

Галереи базы данных векторов признаков

В базе данных векторов признаков на основе Tarantool 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 автомобилей.
- `ffsec_car_objects`: векторы признаков, извлеченные из загруженных в картотеку изображений автомобилей.
- `ffsec_car_clusters`: центроиды кластеров автомобилей.
- `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.

2.9.5 Опции резервного копирования базы данных

Для резервного копирования базы данных векторов признаков вам понадобится утилита `findface-storage-api-dump`. Данная утилита может быть запущена со следующими опциями:

Примечание: Вы можете найти подробную информацию по использованию `findface-storage-api-dump` в разделе *Резервное копирование и восстановление FindFace Multi и всех данных*.

```
findface-storage-api-dump --help
...
Command line flags:
-cache string
    Cache type: inmemory, redis or memcache (default "memcache")
-cache-inmemory-size int
    Maximum number of items in ARC cache (default 16384)
-cache-memcache-dns-cache-timeout duration
    DNS cache timeout (default 1m0s)
-cache-memcache-nodes value
    Comma-separated list of memcache shards (default 127.0.0.1:11211)
-cache-memcache-timeout duration
    Specifies read/write timeout (default 100ms)
```

(continues on next page)

(продолжение с предыдущей страницы)

```

-cache-redis-db int
    Database to be selected after connecting to the server.
-cache-redis-network string
    Network type, either tcp or unix (default "tcp")
-cache-redis-nodes value
    Array of Host:Port addresses (default localhost:6379)
-cache-redis-password string
    Optional password. Must match the password specified in the requirepass_
server configuration option.
-cache-redis-timeout duration
    Specifies dial/read/write timeout (default 5s)
-config string
    Path to config file
-config-template
    Output config template and exit
-continue-on-errors
    Continue on errors instead of exiting
-cpu-profile string
    Enable CPU profile and set output file
-debug
    Enable debug logging
-extraction-api-extraction-api string
    Extraction API address (default "http://127.0.0.1:18666")
-extraction-api-keepalive duration
    keep-alive connection timeout (default 24h0m0s)
-extraction-api-max-idle-conns-per-host int
    max idle keep-alive connections per host (default 20)
-extraction-api-timeouts-connect duration
    extraction-api-timeouts-connect (default 5s)
-extraction-api-timeouts-idle-connection duration
    extraction-api-timeouts-idle-connection (default 10s)
-extraction-api-timeouts-overall duration
    extraction-api-timeouts-overall (default 35s)
-extraction-api-timeouts-response-header duration
    extraction-api-timeouts-response-header (default 30s)
-extraction-api-trace
    Enable HTTP tracing (extremely verbose, slows everything down considerably)
-help
    Print help information
-limits-allow-return-facen
    Allow returning raw feature vectors to detect responses if ?return_
facen=true (v2) or ?return_emben=true (v3)
-limits-body-image-length int
    Maximum length of image supplied in request body (default 33554432)
-limits-deny-networks string
    Comma-separated list of subnets that are not allowed to fetch from (default
"127.0.0.0/8,192.168.0.0/16,10.0.0.0/8,::1/128,fe00::/8")
-limits-url-length int
    Maximum supported url length in bytes (default 4096)
-listen string
    IP:port to listen on (default ":18411")
-normalized-storage-enabled

```

(continues on next page)

(продолжение с предыдущей страницы)

```

        Enables normalize saving (default true)
-normalized-storage-s3-access-key string
        Access key for the object storage
-normalized-storage-s3-bucket-name string
        S3 storage bucket name
-normalized-storage-s3-endpoint string
        S3 compatible object storage endpoint
-normalized-storage-s3-operation-timeout int
        Storage operations (Get,Put,Delete) timeout in seconds (default 30)
-normalized-storage-s3-public-url string
        Storage public url
-normalized-storage-s3-region string
        Storage region
-normalized-storage-s3-secret-access-key string
        Secret key for the object storage
-normalized-storage-s3-secure
        If 'true' API requests will be secure (HTTPS), and insecure (HTTP)
↪ otherwise (default true)
-normalized-storage-webdav-keepalive duration
        keep-alive connection timeout (default 24h0m0s)
-normalized-storage-webdav-max-idle-conns-per-host int
        max idle keep-alive connections per host (default 20)
-normalized-storage-webdav-timeouts-connect duration
        normalized-storage-webdav-timeouts-connect (default 5s)
-normalized-storage-webdav-timeouts-idle-connection duration
        normalized-storage-webdav-timeouts-idle-connection (default 10s)
-normalized-storage-webdav-timeouts-overall duration
        normalized-storage-webdav-timeouts-overall (default 35s)
-normalized-storage-webdav-timeouts-response-header duration
        normalized-storage-webdav-timeouts-response-header (default 30s)
-normalized-storage-webdav-trace
        Enable HTTP tracing (extremely verbose, slows everything down considerably)
-normalized-storage-webdav-upload-url string
        webdav storage for normalized, disable normalized if empty string (default
↪ "http://127.0.0.1:3333/uploads/")
-normalized_storage string
        Normalized storage type: webdav, s3 (default "webdav")
-objects value
        Supported object types (default face,body,car)
-output-dir string
        Output directory (default ".")
-storage-api-cooldown duration
        Cooldown timeout after communication error (default 2s)
-storage-api-galleries-read-slave-first
        Prefer slaves over master for get/list galleries requests
-storage-api-keepalive duration
        keep-alive connection timeout (default 24h0m0s)
-storage-api-max-idle-conns-per-host int
        max idle keep-alive connections per host (default 20)
-storage-api-max-slave-attempts int
        Give up after trying to read from max_slave_attempts slaves (default 2)
-storage-api-read-slave-first

```

(continues on next page)

(продолжение с предыдущей страницы)

```

        Prefer slaves over master for requests
-storage-api-read-slave-only
        Ignore master on read requests. If true: ReadSlaveFirst will be ignored
-storage-api-timeouts-connect duration
        storage-api-timeouts-connect (default 5s)
-storage-api-timeouts-idle-connection duration
        storage-api-timeouts-idle-connection (default 10s)
-storage-api-timeouts-overall duration
        storage-api-timeouts-overall (default 35s)
-storage-api-timeouts-response-header duration
        storage-api-timeouts-response-header (default 30s)
-storage-api-trace
        Enable HTTP tracing (extremely verbose, slows everything down considerably)
...

```

2.9.6 Опции восстановления базы данных

Для восстановления базы данных векторов признаков из резервной копии вам понадобится утилита `findface-storage-api-restore`. Данная утилита может быть запущена со следующими опциями:

Примечание: Вы можете найти подробную информацию по использованию `findface-storage-api-restore` в разделе *Резервное копирование и восстановление FindFace Multi и всех данных*.

```

findface-storage-api-restore --help
...
Command line flags:
-cache string
        Cache type: inmemory, redis or memcache (default "memcache")
-cache-inmemory-size int
        Maximum number of items in ARC cache (default 16384)
-cache-memcache-dns-cache-timeout duration
        DNS cache timeout (default 1m0s)
-cache-memcache-nodes value
        Comma-separated list of memcache shards (default 127.0.0.1:11211)
-cache-memcache-timeout duration
        Specifies read/write timeout (default 100ms)
-cache-redis-db int
        Database to be selected after connecting to the server.
-cache-redis-network string
        Network type, either tcp or unix (default "tcp")
-cache-redis-nodes value
        Array of Host:Port addresses (default localhost:6379)
-cache-redis-password string
        Optional password. Must match the password specified in the requirepass server_
↪ configuration option.
-cache-redis-timeout duration

```

(continues on next page)

(продолжение с предыдущей страницы)

```

    Specifies dial/read/write timeout (default 5s)
-config string
    Path to config file
-config-template
    Output config template and exit
-cpu-profile string
    Enable CPU profile and set output file
-debug
    Enable debug logging
-dont-create-gallery
    Don't create gallery, fail if doesn't exist
-extraction-api-extraction-api string
    Extraction API address (default "http://127.0.0.1:18666")
-extraction-api-keepalive duration
    keep-alive connection timeout (default 24h0m0s)
-extraction-api-max-idle-conns-per-host int
    max idle keep-alive connections per host (default 20)
-extraction-api-timeouts-connect duration
    extraction-api-timeouts-connect (default 5s)
-extraction-api-timeouts-idle-connection duration
    extraction-api-timeouts-idle-connection (default 10s)
-extraction-api-timeouts-overall duration
    extraction-api-timeouts-overall (default 35s)
-extraction-api-timeouts-response-header duration
    extraction-api-timeouts-response-header (default 30s)
-extraction-api-trace
    Enable HTTP tracing (extremely verbose, slows everything down considerably)
-help
    Print help information
-limits-allow-return-facen
    Allow returning raw feature vectors to detect responses if ?return_facen=true
↪ (v2) or ?return_emben=true (v3)
-limits-body-image-length int
    Maximum length of image supplied in request body (default 33554432)
-limits-deny-networks string
    Comma-separated list of subnets that are not allowed to fetch from (default "127.
↪ 0.0.0/8,192.168.0.0/16,10.0.0.0/8,::1/128,fe00::/8")
-limits-url-length int
    Maximum supported url length in bytes (default 4096)
-listen string
    IP:port to listen on (default ":18411")
-normalized-storage-enabled
    Enables normalize saving (default true)
-normalized-storage-s3-access-key string
    Access key for the object storage
-normalized-storage-s3-bucket-name string
    S3 storage bucket name
-normalized-storage-s3-endpoint string
    S3 compatible object storage endpoint
-normalized-storage-s3-operation-timeout int
    Storage operations (Get,Put,Delete) timeout in seconds (default 30)
-normalized-storage-s3-public-url string

```

(continues on next page)

(продолжение с предыдущей страницы)

```

Storage public url
-normalized-storage-s3-region string
    Storage region
-normalized-storage-s3-secret-access-key string
    Secret key for the object storage
-normalized-storage-s3-secure
    If 'true' API requests will be secure (HTTPS), and insecure (HTTP) otherwise
↪(default true)
-normalized-storage-webdav-keepalive duration
    keep-alive connection timeout (default 24h0m0s)
-normalized-storage-webdav-max-idle-conns-per-host int
    max idle keep-alive connections per host (default 20)
-normalized-storage-webdav-timeouts-connect duration
    normalized-storage-webdav-timeouts-connect (default 5s)
-normalized-storage-webdav-timeouts-idle-connection duration
    normalized-storage-webdav-timeouts-idle-connection (default 10s)
-normalized-storage-webdav-timeouts-overall duration
    normalized-storage-webdav-timeouts-overall (default 35s)
-normalized-storage-webdav-timeouts-response-header duration
    normalized-storage-webdav-timeouts-response-header (default 30s)
-normalized-storage-webdav-trace
    Enable HTTP tracing (extremely verbose, slows everything down considerably)
-normalized-storage-webdav-upload-url string
    webdav storage for normalized, disable normalized if empty string (default
↪"http://127.0.0.1:3333/uploads/")
-normalized_storage string
    Normalized storage type: webdav, s3 (default "webdav")
-objects value
    Supported object types (default face,body,car)
-rename string
    Ignore dump header and use this string as gallery name
-storage-api-cooldown duration
    Cooldown timeout after communication error (default 2s)
-storage-api-galleries-read-slave-first
    Prefer slaves over master for get/list galleries requests
-storage-api-keepalive duration
    keep-alive connection timeout (default 24h0m0s)
-storage-api-max-idle-conns-per-host int
    max idle keep-alive connections per host (default 20)
-storage-api-max-slave-attempts int
    Give up after trying to read from max_slave_attempts slaves (default 2)
-storage-api-read-slave-first
    Prefer slaves over master for requests
-storage-api-read-slave-only
    Ignore master on read requests. If true: ReadSlaveFirst will be ignored
-storage-api-timeouts-connect duration
    storage-api-timeouts-connect (default 5s)
-storage-api-timeouts-idle-connection duration
    storage-api-timeouts-idle-connection (default 10s)
-storage-api-timeouts-overall duration
    storage-api-timeouts-overall (default 35s)
-storage-api-timeouts-response-header duration

```

(continues on next page)

(продолжение с предыдущей страницы)

```

storage-api-timeouts-response-header (default 30s)
-storage-api-trace
  Enable HTTP tracing (extremely verbose, slows everything down considerably)
...

```

2.9.7 Прямые API-запросы к базе данных Tarantool

Вы можете использовать HTTP API для извлечения данных объектов (лиц, силуэтов, автомобилей) напрямую из базы данных Tarantool.

Примечание: В текущей реализации API Tarantool оперирует объектами как лицами. Например, для добавления объекта нужно отправить запрос POST `/:ver/faces/add/:name`.

В этом разделе:

- *Общие сведения*
- *Добавление объекта*
- *Удаление объекта*
- *Поиск объектов*
- *Редактирование метаданных и/или вектора признаков объекта*
- *Получение списка галерей*
- *Получение информации о галерее*
- *Создание галереи*
- *Удаление галереи*

Общие сведения

API-запросы к базе данных Tarantool нужно отправлять по адресу `http://<tarantool_host_ip:port>`.

Совет: Порт для API-запросов можно узнать в разделе `FindFace.start` файла конфигурации Tarantool `/etc/tarantool/instances.available/*.lua`:

```

cat /etc/tarantool/instances.available/*.lua

##8101:
FindFace.start("127.0.0.1", 8101)

```

Примечание: В случае если FindFace Multi развернут на одиночном физическом сервере, база данных Tarantool по умолчанию будет доступна только локально (127.0.0.1). Если необходимо открыть

доступ к базе данных Tarantool с удаленного сервера, *внесите изменения* в файл конфигурации `findface-tarantool-server (/etc/tarantool/instances.available/*.lua)`.

API-запросы к Tarantool могут содержать следующие параметры в сегментах пути:

- `:ver`: версия API (v2 на данный момент).
- `:name`: имя галереи.

По умолчанию в базе данных Tarantool созданы 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 автомобилей.
- `ffsec_car_objects`: векторы признаков, извлеченные из загруженных в картотеку изображений автомобилей.
- `ffsec_car_clusters`: центроиды кластеров автомобилей.
- `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.

Совет: Для получения списка имен галерей на шарде введите следующую команду в адресном поле браузера:

```
http://<tarantool_host_ip:shard_port>/stat/list/1/99
```

Та же самая команда в консоли:

```
curl <tarantool_host_ip:shard_port>/stat/list/1/99 \ | jq
```

Вы также можете получить список имен галерей, отправив в Tarantool прямой запрос:

```
echo 'box.space.galleries:select()' | tarantoolctl connect <tarantool_host_ip:shard_port>
```

Имейте в виду, что при значительном количестве шардов в системе произвольно выбранный шард может не включать в себя все существующие галереи. В этом случае отобразите список галерей на нескольких шардах.

Добавление объекта

```
POST /:ver/faces/add/:name
```

Параметры в теле:

Массив объектов в представлении JSON со следующими полями:

- "id": id объекта в галерее, uint64_t,
- "facen": необработанный вектор признаков, base64,
- "meta": метаданные объекта, словарь.

Возвращает:

- HTTP 200 и пустое тело в случае успеха.
- HTTP 404 с описанием ошибки, если галерея с заданным именем не существует.
- HTTP с отличным от 200 статусом и описанием ошибки в теле в случае неудачи.

Пример

Запрос

```
curl -D - -s 'http://localhost:8101/v2/faces/add/testgal' --data '
[
  {
    "id": 9223372036854776000,
    "facen": "qgI3vZRv/z...Np09MdHavW1WuT0=",
    "meta": {
      "cam_id": "223900",
      "person_name": "Mary Ostin",
    }
  }
]
```

Ответ

```
HTTP/1.1 200 Ok
Content-length: 1234
Server: Tarantool http (tarantool v1.7.3-673-g23cc4dc)
Connection: keep-alive
```

Удаление объекта

```
POST /v2/faces/delete/:name
```

Параметры в теле:

Массив в представлении JSON из списка id объектов, подлежащих удалению

Возвращает:

- HTTP 200 и пустое тело в случае успеха.
- HTTP 404, если объект с заданным id не найден в галерее.
- HTTP с отличным от 200 статусом и описание ошибки в теле в случае неудачи.

Пример

Запрос

```
curl -D - -s 'http://localhost:8101/v2/faces/delete/testgal' --data '[1, 4, 922, 3]'
```

Ответ

```
HTTP/1.1 200 Ok
Content-length: 111
Server: Tarantool http (tarantool v1.7.3-673-g23cc4dc)
Connection: keep-alive
```

Поиск объектов

```
POST /v2/faces/search/:name
```

Параметры в теле:

Поисковый запрос в представлении JSON со следующими полями:

- **limit**: максимальное количество объектов в ответе.
- **sort**: включает сортировку по следующим параметрам: **id**: по возрастанию id, **-id** по убыванию id, **-score**: по убыванию степени схожести (если поиск выполняется по схожим векторам признаков).
- **filter** (фильтры):
 - **facen**: опциональный фильтр по схожести вектора признаков. Передайте словарь со следующими полями: **data**: вектор признаков в формате base64; **score**: диапазон схожести объектов [пороговая схожесть; 1], поддерживаются только запросы с правой границей 1 (сто процентное совпадение объектов).

– `id` и `meta/<meta_key>`: фильтры по пользовательскому `id` объектов и содержимому поля `meta`. Для задания фильтра используются следующие операторы:

- * `range`: диапазон значений, только для числовых полей.
- * `set`: набор значений, одно из которых должно присутствовать в `id` или метаданных, для числовых и строковых полей.
- * `subset`: набор значений, каждое из которых должно присутствовать в `id` или метаданных, для числовых и строковых полей.
- * `like`: аналогично `like` в SQL-запросах: поддерживаются только `'aa%'` или `'aa%'` или `'%aa%'`. Только для полей `string` и `set[string]`. При использовании `set[string]` фильтр вернет результат, если хотя бы одно из значений прошло проверку.
- * `ilike` (только для полей `string` и `set[string]`): аналогично `like`, но без учета регистра.

Возвращает:

- В случае успеха массив JSON с объектами. Значение в заголовке `X-search-stat` показывает, был ли использован быстрый индекс для поиска: `with_index` или `without_index`.

Примечание: В API v2 быстрый индекс не используется.

- HTTP с отличным от 200 статусом и описание ошибки в теле в случае неудачи.

Пример

Запрос

```
curl -D - -s 'http://localhost:8101/v2/testgal/search' --data '
{
  "limit": 2,
  "sort": {
    "score": -1
  },
  "filter": {
    "facen": {
      "data": "qgI3vZRv/zOBQTk9rcir0yZrNp09MdHavW1WuT0=",
      "score": [0.75, 1]
    },
    "id": {
      "range": [9223372036854000000, 9223372036854999000]
    },
    "meta": {
      "person_id": {
        "range": [444, 999]
      },
      "cam_id": {
        "set": ["12767", "8632", "23989"]
      }
    }
  }
}
```

(continues on next page)

(продолжение с предыдущей страницы)

```
}  
'
```

Ответ

```
HTTP/1.1 200 Ok  
Content-length: 1234  
X-search-stat: without_index  
Server: Tarantool http (tarantool v1.7.3-673-g23cc4dc)  
Connection: keep-alive  
  
{  
  "results": [  
    {  
      "facen": " qgI3vZRv/zOBQTk9rcir0yZrNp09MdHavW1WuT0=",  
      "meta": {  
        "timestamp": 0,  
        "photo_hash": "",  
        "person_id": 777,  
        "cam_id": "8632"  
      },  
      "score": 0.9964,  
      "id": 9223372036854776000  
    }  
  ]  
}
```

Редактирование метаданных и/или вектора признаков объекта

```
POST /v2/faces/update/:name
```

Параметры в теле:

Массив объектов в представлении JSON со следующими полями:

- "id": id объекта, uint64_t.
- "facen": (опционально) новый вектор признаков, base64. Если параметр отсутствует или null, поле в базе данных не обновляется.
- "meta": словарь, в котором передаются новые метаданные. Если поле meta отсутствует или null, оно не обновляется в базе данных.

Возвращает:

- HTTP 200 и словарь со всеми параметрами объекта, в том числе неизменными, в случае успеха.
- HTTP 404 с описанием ошибки, если объекта с таким id не существует.
- HTTP с отличным от 200 статусом и описанием ошибки в теле в случае неудачи.

Пример**Запрос**

```
curl -D - -s 'http://localhost:8101/v2/faces/update/sandbox' --data ' [{"id":1,"facen
↪":null,"meta":{"m:timestamp":1848}}] '
```

Ответ

```
HTTP/1.1 200 Ok
Content-length: 151
Server: Tarantool http (tarantool v1.7.3-673-g23cc4dc)
Connection: keep-alive

{"meta":{"m:timestamp":1848,"normalized_id":"1_b9pkrf00mjt6h1vmq1kg.png","m:cam_id":
↪"a9f7a973-f07e-469d-a3bd-41ddd510b26f","feat":{"score":0.123}}, "id":1, ... }
```

Получение списка галерей

```
POST /v2/galleries/list
```

Возвращает:

Массив с галереями, для каждой из которой возвращается имя (`name`) и количество объектов (`faces`).

Пример**Запрос**

```
curl -D - -s -X POST http://localhost:8101/v2/galleries/list
```

Ответ

```
HTTP/1.1 200 Ok
Content-length: 42
Server: Tarantool http (tarantool v1.7.3-673-g23cc4dc)
Connection: keep-alive

{
  "results": [
    {
      "name": "testgal",
      "faces": 2
    }
  ]
}
```

Получение информации о галерее

```
POST /v2/galleries/get/:name
```

Возвращает:

- HTTP 200 и словарь с параметрами галереи в случае успеха.
- HTTP 404 с описанием ошибки, если галереи с таким именем не существует.
- HTTP с отличным от 200 статусом и описанием ошибки в теле в случае неудачи.

Пример

Запрос

```
curl -D - -s -X POST http://localhost:8101/v2/galleries/get/testgal
```

```
HTTP/1.1 200 Ok
Content-length: 11
Server: Tarantool http (tarantool v1.7.3-673-g23cc4dc)
Connection: keep-alive

{"faces":2}
```


Создание галереи

```
POST /v2/galleries/add/:name
```

Возвращает:

- HTTP 200 и пустое тело в случае успеха.
- HTTP с отличным от 200 статусом и описание ошибки в теле в случае неудачи.

Пример

Запрос

```
curl -D - -X POST -s 'http://localhost:8101/v2/galleries/add/123'
```

Ответ

```
HTTP/1.1 200 Ok
Content-length: 111
Server: Tarantool http (tarantool v1.7.3-673-g23cc4dc)
Connection: keep-alive
```

Удаление галереи

```
POST /v2/galleries/delete/:name
```

Возвращает:

- HTTP 200 и пустое тело в случае успеха.
- HTTP с отличным от 200 статусом и описание ошибки в теле в случае неудачи.

Пример

Запрос

```
curl -D - -X POST -s 'http://localhost:8101/v2/galleries/delete/123'
```

Ответ

```
HTTP/1.1 204 No content
Content-length: 0
Server: Tarantool http (tarantool v1.7.3-673-g23cc4dc)
Connection: keep-alive
```

В этой главе описывается, как работать с веб-интерфейсом FindFace Multi, включая его расширенные возможности. Данные сведения будут интересны администраторам, аналитикам, 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 автомобилей. Измерение расстояния между людьми*
- *Управление областями*
- *Анализ людей*
- *Работа в штатном режиме*
- *Базовое обслуживание системы*
- *Больше возможностей*

3.1.1 Подготовка к работе

Выполните первичную настройку системы:

1. *Настройте* левую панель навигации.
2. *Выберите* язык.

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. *Включите видеозапись для выбранных камер* (если активирована функция Видеомегнитофон).

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, автомобилей) 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:

1. *Настройте содержимое карточки*. Создайте дополнительные поля, вкладки и поисковые фильтры.
2. *Создайте связи между карточками*, чтобы наилучшим образом использовать аналитические возможности своей системы.

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 автомобиля.
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:

1. Для расширенного функционала видеонаблюдения используйте *Видеомagneитофон*.

3.1.6 Подсчет людей и автомобилей. Измерение расстояния между людьми

Настройте *счетчики* для подсчета лиц, силуэтов и автомобилей 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 автомобильных пробок и т. д.

3.1.7 Управление областями

Создайте *области*, чтобы контролировать присутствие в них людей по заданным правилам и расписаниям. Возможные варианты использования: предотвращение длинных очередей в торговых точках, предотвращение краж на предприятиях в нерабочее время, контроль опасных зон, учет рабочего времени.

3.1.8 Анализ людей

FindFace Multi предоставляет набор инструментов для анализа людей:

1. *Анализируйте* социальные взаимодействия. Изучите круг людей, с которыми человек контактировал.
2. Задействуйте аналитику ‘Знай своего клиента’ (KYC). Это аналитика по количеству посетителей, их полу, среднему возрасту, наиболее часто посещаемым зонам и характеру посещений (первое посещение или возвращение). *Подробнее*.

3.1.9 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 автомобиля), 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, аналитику КУС, журнал действий пользователей, области.

3.1.10 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*.

3.1.11 Больше возможностей

1. Настройте *вебхуки* для автоматической отправки уведомлений об определенных событиях, эпизодах, записях счетчика и областях 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*.
3. Ознакомьтесь со списком наших *партнерских интеграций*.
4. Интегрируйте *периферийное устройство*.

См.также:

- *Управление пользователями и безопасность системы*
- *Управление видеокамерами. Видеомагнитофон*
- *Настройка мониторинга объектов и картотеки*
- *Идентификация объектов в офлайн видео*
- *Счетчики лиц, силуэтов и машин. Определение дистанции*

- *Кластеры лиц, силуэтов, автомобилей*
- *Связанная с людьми аналитика*
- *Настройка расширенного функционала*
- *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.

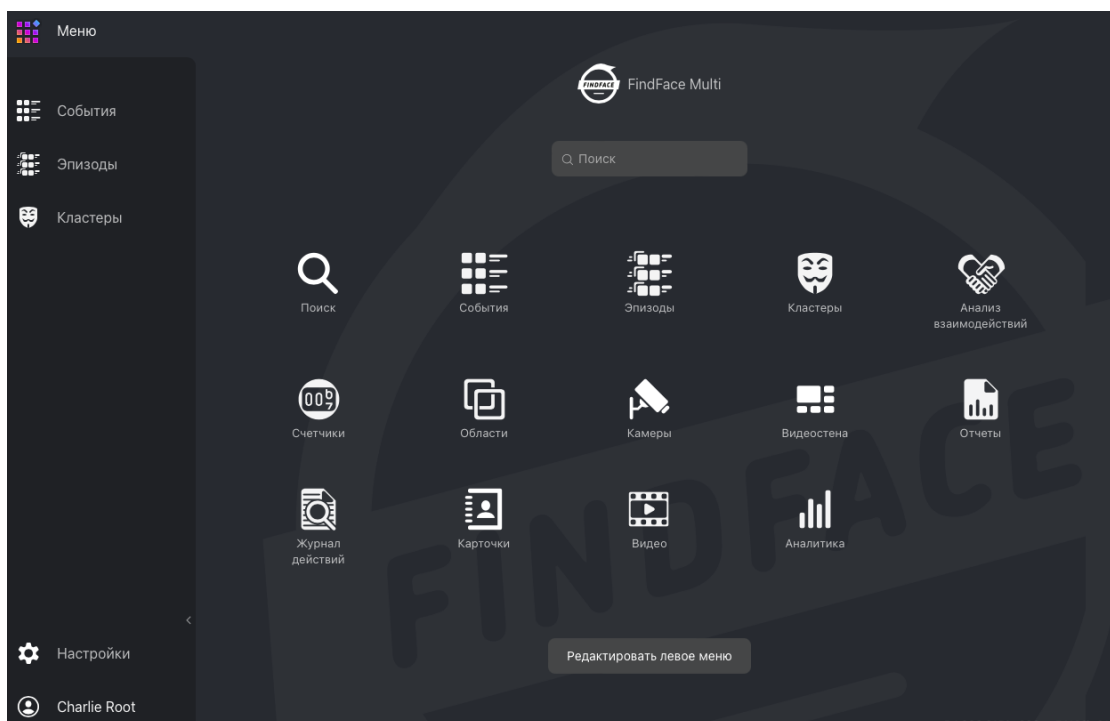
Примечание: Базовый адрес задается при *установке* FindFace Multi.

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

По умолчанию на левой панели навигации есть только две вкладки: *События* и *Эпизоды*. Для доступа к другим вкладкам FindFace Multi используйте *Меню*.

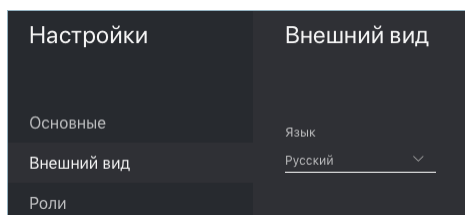


С помощью *Меню* вы также сможете сделать свои любимые вкладки постоянно доступными из панели навигации. Выполните следующие действия:

1. Нажмите *Редактировать левое меню*.
2. Установите/снимите флажки с нужных вкладок.
3. Нажмите *Завершить редактирование*.

3.2.2 Язык веб-интерфейса

Для того чтобы переключить язык системы, перейдите в меню *Настройки* -> *Внешний вид*.



Примечание: По умолчанию в веб-интерфейсе для выбора доступны только английский и русский языки. Другие языки предоставляются по запросу (support@ntechlab.com) и добавляются в систему через файл конфигурации `/etc/findface-security/config.py`.

См.также:

Добавление пользовательского языка

3.3 Управление видеокameraми. Видеоманитофон

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 пользователями*).

В этой главе:

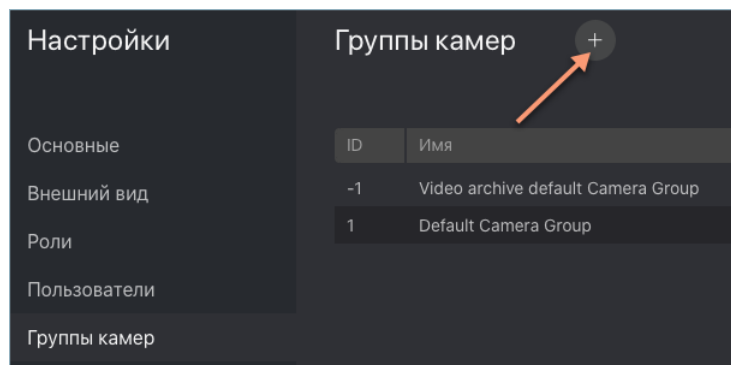
- *Создание группы камер*
- *Добавление камеры*
- *Мониторинг работы камер*
- *Видеоманитофон*
 - *Включение видеозаписи с камеры*
 - *Просмотр видео с камеры в видеоплеере*

3.3.1 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 *+*.



3. На вкладке *Информация*, введите имя группы и при необходимости комментарий к ней.

Создать группу камер

Информация

Разрешения

Имя

Офис

Комментарий

Метки

Введите или выберите метки

Дедуплицировать события

☒

Фиксировать только уникальные события среди камер группы, исключив дубликаты.

Интервал дедупликации

15

Интервал в секундах, с которым события проверяются на уникальность.

Порог срабатывания

☐

Изменение данного параметра повлияет на работоспособность системы. Не трогайте, если вы не знаете точно, что делаете

☒ Активная

Сохранить

Назад

- 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 `/etc/findface-video-worker-gpu.ini` (`/etc/findface-video-worker-cpu.ini`). Подробнее см. *Привязка группы камер к экземпляру findface-video-worker*.

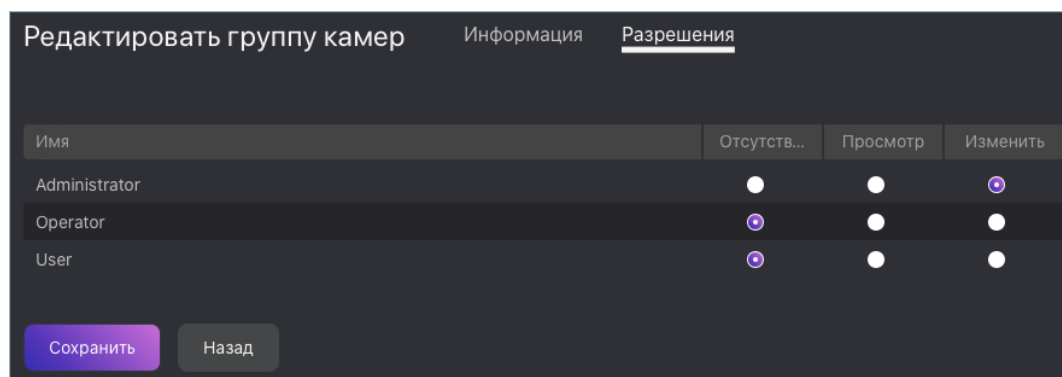
- Если события от камер, принадлежащих одной группе, требуется дедуплицировать, т. е. исключить одинаковые события, поставьте флажок *Дедуплицировать события* и задайте в секундах интервал дедупликации (интервал, с которым события проверяются на уникальность).

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

- 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).

- Поставьте флажок *Активная*.
- Нажмите *Сохранить*.
- На вкладке *Разрешения* назначьте права на работу с группой камер, указав, пользователям с какими ролями разрешено изменять/просматривать ее настройки.

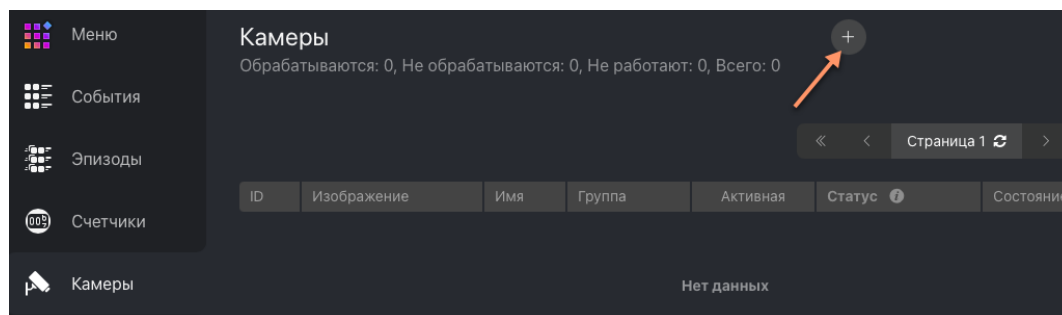


- Нажмите *Сохранить*.

3.3.2 Добавление камеры

Для добавления камеры в группу выполните следующие действия:

- Перейдите на вкладку *Камеры*.
- Нажмите *+*.



- На вкладке *Настройки* -> *Основное* введите общие данные камеры:

- Укажите имя камеры.
- Добавьте камеру в группу.
- Укажите URL камеры (*Видеопоток*). Если вы используете камеру ONVIF, выберите ее из списка обнаруженных устройств для автоматической загрузки доступных настроек и видеопотоков.

Примечание: Вы также можете использовать объект типа камера для интеграции периферийного устройства. [Подробнее](#).

- При необходимости добавьте комментарий.
- Поставьте флажок *Активная*.

4. На вкладке *Настройки* -> *Дополнительное* выполните тонкую настройку камеры:

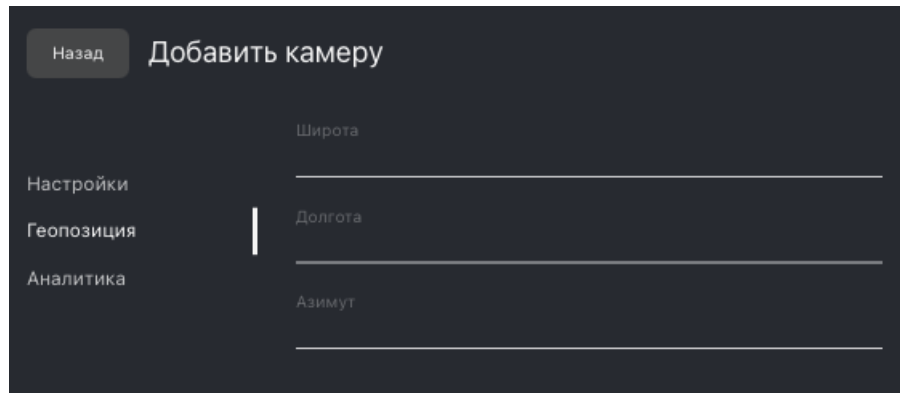
- Поставьте флажки для тех детекторов, которые нужно активировать для данной камеры: лица, силуэты, автомобили.
- При необходимости измените ориентацию видео.

Важно: Вращение потока выполняется на стороне сервера `findface-security` средствами постобработки. Это может негативно сказаться на производительности. Рекомендуем по возможности делать вращение потока на стороне камеры.

- *Время ожидания ответа на запрос:* Таймаут в миллисекундах отправки на сервер обнаруженных объектов.
- *Получать временные метки из потока:* Поставьте флажок, чтобы отправлять на сервер временные метки, полученные из потока. Снимите флажок, чтобы отправлять текущие дату и время.
- *Прибавлять к временным меткам:* Прибавлять указанное количество секунд к временным меткам из потока.
- *Формат FFMPEG:* Передать формат FFMPEG (`mxg`, `flv`, и т. д.), если он не может быть автоматически определен.
- *Опции FFMPEG:* Опции FFMPEG для видеопотока. Задаются массивом строк `ключ=значение`, например, `"rtsp_transport=tcp ss=00:20:00"`.
- *Минимальная интенсивность движения:* Минимальная интенсивность движения, которая будет регистрироваться детектором движения.

- *Проверить SSL-сертификат*: Поставьте флажок, чтобы проверять SSL-сертификат сервера при отправке на него объектов через https. Снимите флажок, если вы используете самоподписанный сертификат.

5. (Опционально) На вкладке *Геопозиция* укажите географические координаты камеры.



6. На вкладке *Аналитика* укажите настройки детектора объектов каждого типа.

Назад
Добавить камеру

Настройки
Геопозиция
Аналитика

Лица
Силуэты
Автомобили

Минимальное качество изображения объекта (filter_min_quality)
0.45 0.45
Минимальное качество изображения лица для детекции.
Допустимый диапазон от 0 до 1.
Базовое рекомендуемое значение 0.45 - соответствует
удовлетворительному качеству изображения лица.
Значения вблизи 1 соответствуют наиболее качественным выровненным
изображениям лиц, полностью находящихся в кадре.
Значения вблизи 0 соответствуют замыленным изображениям, а также
изображениям, содержащим перевернутые лица или лица, повернутые под
большими углами. При таких значениях распознавание может быть
неэффективным.

Минимальный размер объекта (filter_min_size)
1 1
Минимальный размер объекта в пикселях для отправки на сервер.

Максимальный размер объекта (filter_max_size)
8192 8192
Максимальный размер объекта в пикселях для отправки на сервер.

Качество сжатия (jpeg_quality)
95 95
Качество сжатия полного кадра для отправки.

Буферный режим (overall_only)
☐ false
Буферный режим. Отправлять для объекта один кадр наилучшего качества.

Временной интервал (realtime_post_interval)
1 1
Временной интервал в секундах (целое или рациональное число), в течение
которого в режиме реального времени выбирается лучший кадр с объектом.

Отправлять первый объект сразу (realtime_post_first_immediately)
☐ false
Если true, 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. Если false,
отправлять первый объект после окончания первого realtime_post_interval.

Отправлять лучший кадр (realtime_post_every_interval)
☐ false
Если true, отправлять лучший кадр в каждом временном интервале
realtime_post_interval в режиме реального времени. Если false, отправлять
лучший кадр, только если его качество улучшилось по сравнению с
предыдущим отправленным кадром.

Обрезать полный кадр по ROT (fullframe_crop_rot)
☐ false
Если true, то обрезать полный кадр по размеру ROT-области перед
отправкой на распознавание. Размер полного кадра будет равен размеру
ROT-области.

Дополнительные параметры

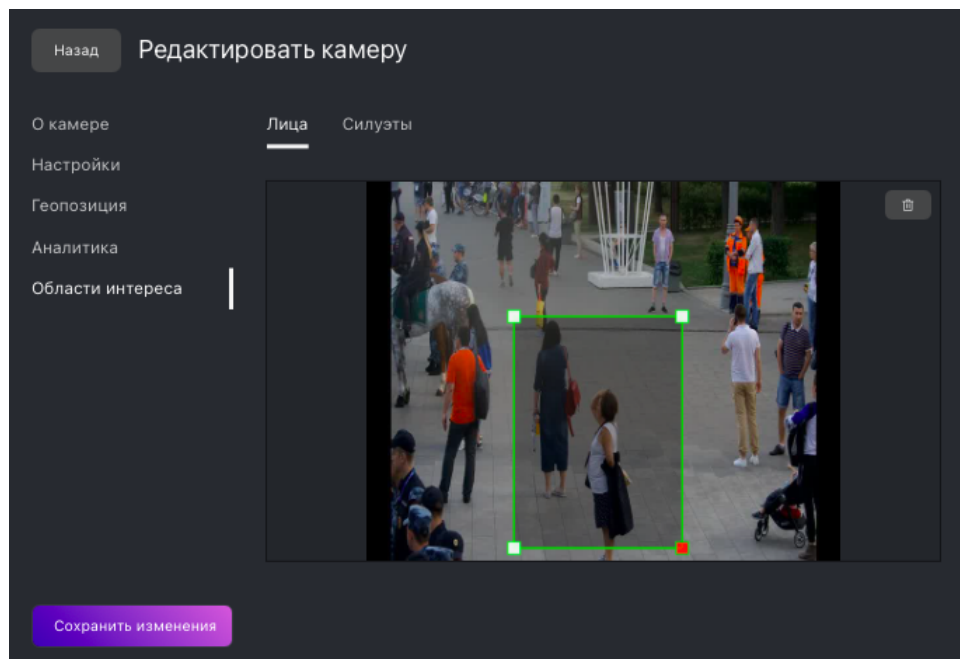
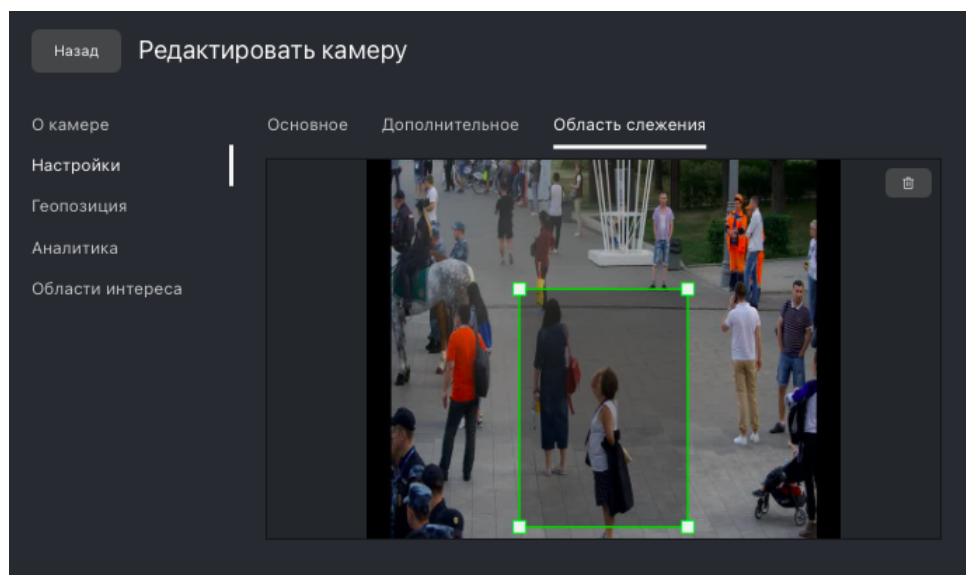
- **Минимальное качество изображения объекта:** Минимальное качество изображения объекта для детекции. Не меняйте значение по умолчанию без предварительной консультации с нашими техническими специалистами (support@ntechlab.com).
- **Минимальный размер объекта:** Минимальный размер объекта в пикселях для отправки на сервер.
- **Максимальный размер объекта:** Максимальный размер объекта в пикселях для отправки на сервер.
- **Качество сжатия:** Качество сжатия полного кадра для отправки.
- **Буферный режим.** Отправлять для объекта один кадр наилучшего качества.
- **Временной интервал:** Временной интервал в миллисекундах (целое или десятичное число), в течение которого в режиме реального времени выбирается лучший кадр с объектом.
- **Отправлять первый объект сразу:** Поставьте флажок, чтобы отправлять первое изобра-

жение объекта из трека сразу, как только фильтры по качеству/размеру/региону интереса пройдены, не дожидаясь окончания первого Временного интервала. Способ отправки последующих изображений объекта зависит от установленного значения Отправлять лучший кадр. Снимите флажок, чтобы отправлять первое изображение объекта из трека только после окончания первого Временного интервала.

- *Отправлять лучший кадр*: Поставьте флажок, чтобы отправлять лучший кадр в каждом Временном интервале в режиме реального времени. Снимите флажок, чтобы отправлять лучший кадр, только если его качество улучшилось по сравнению с предыдущим отправленным кадром.

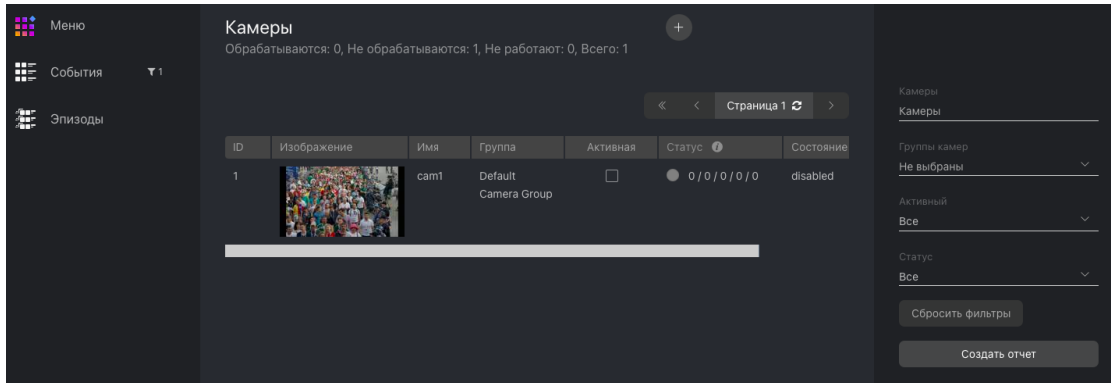
7. Нажмите *Сохранить изменения*.

8. При необходимости укажите регион слежения в поле зрения камеры (*Настройки -> Область слежения*) и зоны детекции (*Области интереса*) для каждого типа объектов. Нажмите *Сохранить изменения*.



3.3.3 Мониторинг работы камер

Мониторинг работы камер выполняется на вкладке *Камеры*.



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 `/etc/findface-security/config.py`:

```
sudo vi /etc/findface-security/config.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_FACES': {'yellow': 0.1, 'red': 0.3},
    ...
}
```

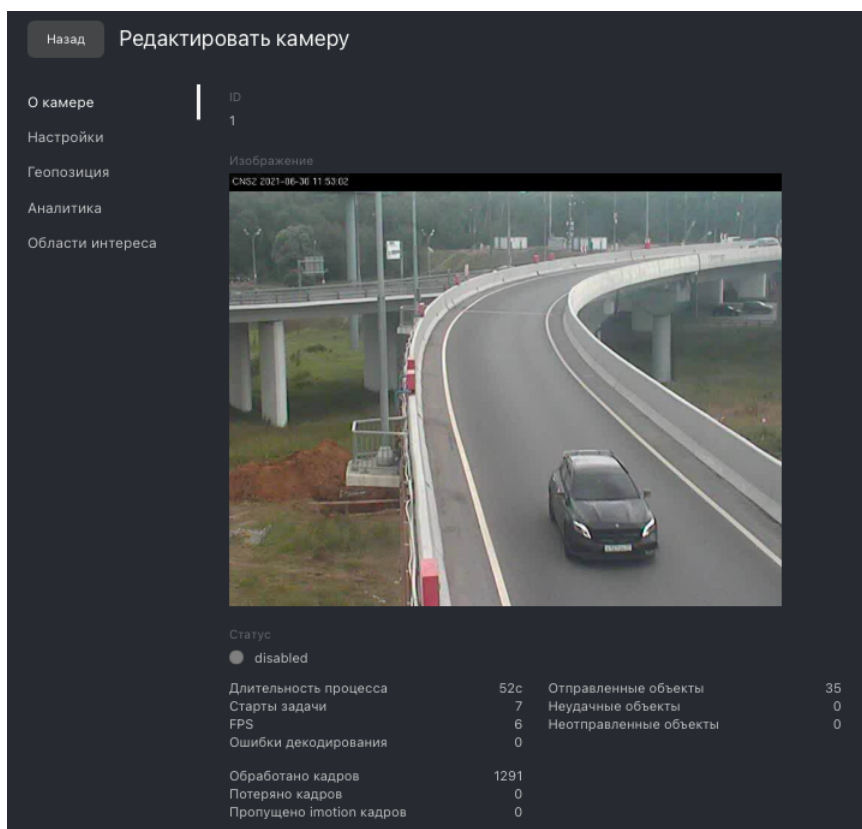
Каждая камера связана с так называемым job-заданием, задачей на обработку видео, содержащей параметры конфигурации и данные видеопотока, которая назначается определенному экземпляру `findface-video-worker`. Данная задача может быть перезапущена.

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:

- *Группы камер*,
- *Активный*,
- *Статус*.

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 *О камере*.



3.3.4 Видеоманитофон

Включение видеозаписи с камеры

Если Видеоманитофон *развернут* и *настроен*, в основных настройках камер появится флажок *Включить запись*. Установите этот флажок, чтобы включить запись видеофрагментов с камеры и отправку их в Видеоманитофон для дальнейшей обработки.

Назад

Редактировать камеру

О камере
Настройки
Геопозиция
Аналитика
Области интереса

Основное

Дополнительное
Область слежения

* Имя
5fl-entrance

* Группа камер
Default Camera Gro...

Тип соединения

Видеопоток

ONVIF
Edge-устройство

* URL видеопотока
rtmp://restreamer.int.ntl/cams/5fl-entrance

Комментарий

☒ Активная

☒ Включить запись

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.

Назад

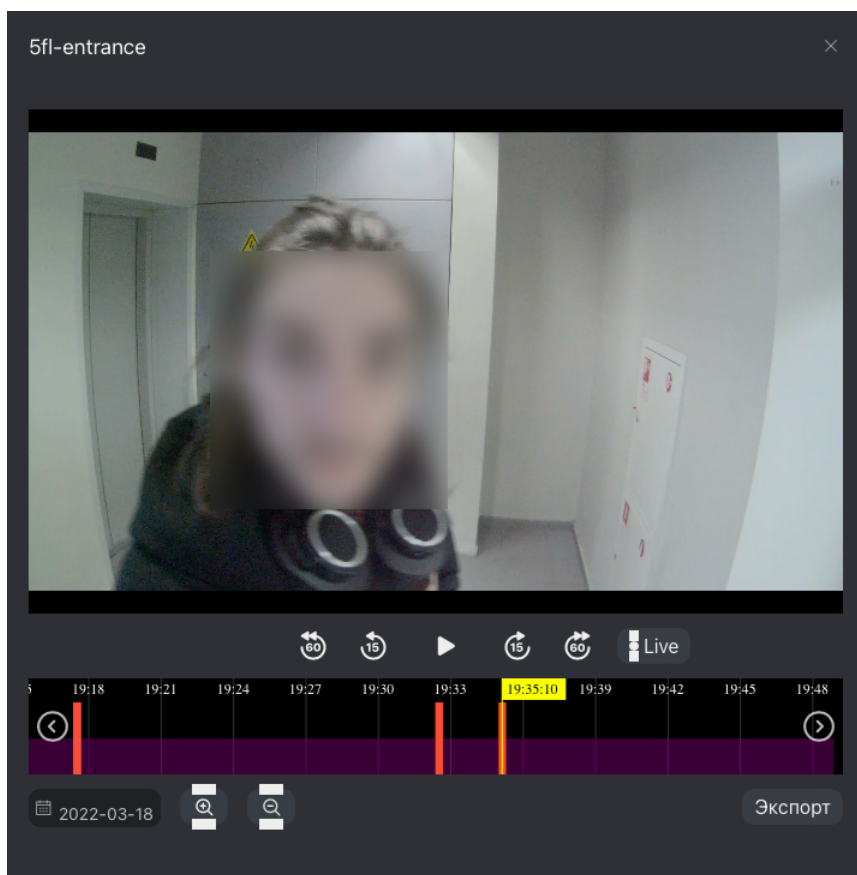
Камеры

Обрабатываются: 2, Не обрабатываются: 0, Не работают: 0, Всего: 2

<<
<
Страница 1
>
>>

ID	Изображение	Имя	Группа	Активная	Статус	Состояние
1		5fl-entrance	Default Camera Group	<input checked="" type="checkbox"/>	49м 21с /28 /0 /0 /11	in progress
2		openspace	Default Camera Group	<input checked="" type="checkbox"/>	49м 15с /259 /0 /0 /12	in progress

<<
<
Страница 1
>
>>



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 автомобилями (если данный функционал активен), в виде цветных маркеров на временной шкале. События без совпадений отмечаются розовым цветом, а с совпадениями — салатовым.

Примечание: Данная функция 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 водя курсором по временной шкале с зажатой правой кнопкой мыши.

Совет: Для увеличения и уменьшения масштаба используйте кнопки *+* / *-* или колесико мыши.

5. Экспорт выбранных клипов.

См.также:

- *Привязка группы камер к экземпляру findface-video-worker*
- *Дедупликация событий*
- *Видеодетекция объектов: findface-video-manager и findface-video-worker*
- *Пошаговое развертывание функционала Видеомегнитофон*
- *Настройка Видеомегнитофона*
- *Видеоплеер в уведомлении о событии*

3.4 Настройка мониторинга объектов и картотеки

FindFace Multi обеспечивает видеомониторинг следующих объектов:

- человеческое лицо
- человеческий силуэт
- автомобиль/транспортное средство

Мониторинг объектов реализован посредством совокупности встроенных и пользовательских списков наблюдения, например, Разыскивается, VIP и т. п., и картотеки. Вы можете создать столько пользовательских списков наблюдения, сколько необходимо.

Картотека содержит два типа карточек:

- карточка человека: содержит данные о человеке, такие как изображения его лица и силуэта, имя и связанные файлы.
- карточка автомобиля: содержит связанные с автомобилем данные, такие как изображения автомобиля и регистрационный номер.

Каждая карточка входит в один или несколько списков наблюдения. Для того чтобы поставить человека/автомобиль на видеомониторинг, необходимо активировать соответствующий список наблюдения.

Для автоматического создания большого количества карточек используйте *функционал пакетной загрузки*.

В этом разделе:

- *Мониторинг незарегистрированных объектов*
- *Создание списка наблюдения*
- *Создание карточки человека*
- *Создание карточки автомобиля*
- *Просмотр сопоставленных кластеров*
- *Связи карточки*

- *Пакетная загрузка карточек*
- *Фильтры карточек*
- *Очистка картотеки*
- *Отключение создания событий для определенных объектов*
- *Автокарточки и заполненные карточки*

3.4.1 Мониторинг незарегистрированных объектов

Базовая конфигурация FindFace Multi уже содержит предустановленный список наблюдения, предназначенный для мониторинга незарегистрированных в системе объектов, т. е. объектов, 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* в таблице.

Редактировать список наблюдения

Информация

Разрешения

Метка

ID

-1

Имя

Unmatched

Группы камер

Не выбраны

Если список групп камер пустой, то список досье работает по всем камерам.

Комментарий

Default list for unmatched events

☐ Требовать подтверждение события

☐ Включить звуковое оповещение

☐ Не создавать события

☒ Активный

Сохранить

Назад

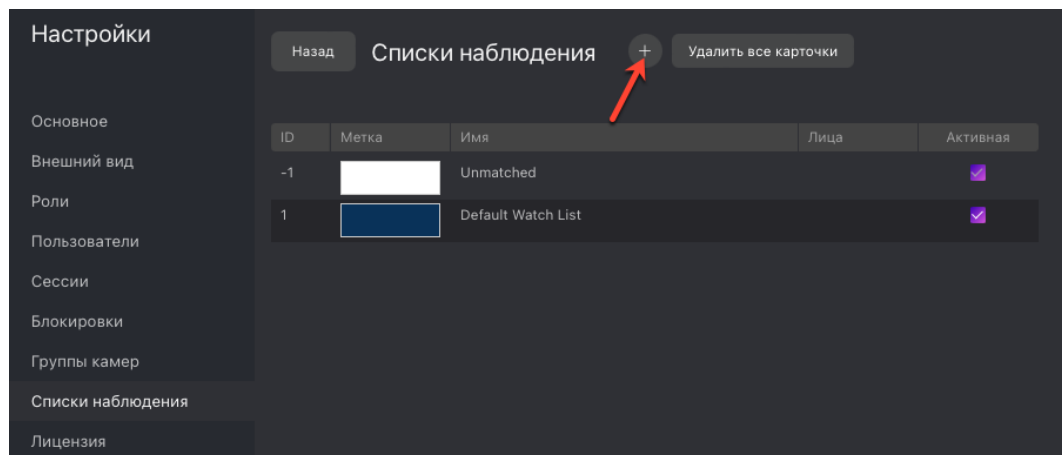
Добавить синхронизацию

Удалить все досье

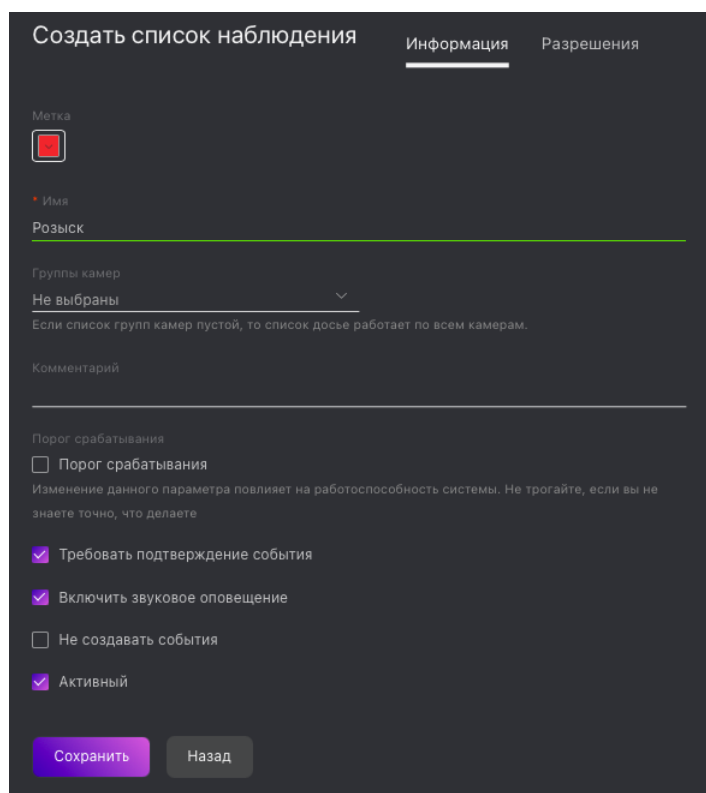
3.4.2 Создание списка наблюдения

Для создания пользовательского списка наблюдения выполните следующие действия:

1. Перейдите на вкладку *Настройки*. Выберите *Списки наблюдения*.
2. Нажмите **+**.



3. В палитре *Метка* выберите цвет, который будет использоваться в оповещениях о событиях распознавания из данного списка. Правильно выбранный цвет повышает быстроту реагирования оператора на событие.

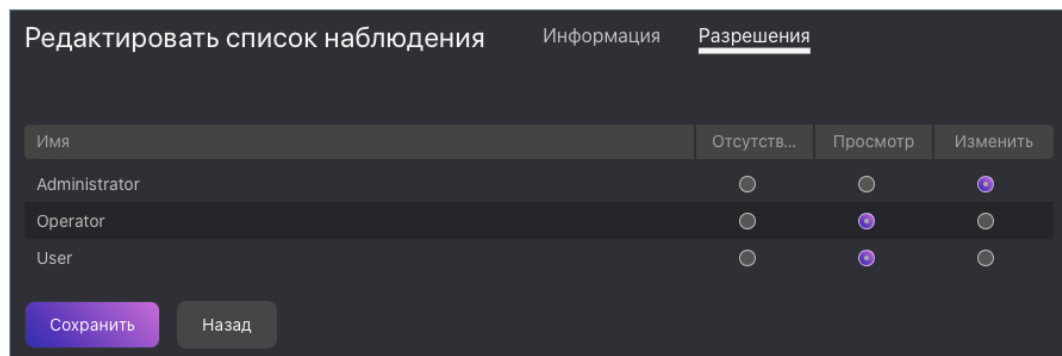


4. Введите название списка наблюдения. При необходимости добавьте комментарий.

5. Выберите группу камер, которая будет использоваться для мониторинга списка наблюдения. Если группа камер не выбрана, список наблюдения будет отслеживаться всеми активными камерами в системе.
6. Поставьте флажок *Требовать подтверждение*, если для данного списка оператор должен в обязательном порядке подтвердить принятие события.
7. При необходимости включите звук при появлении события для данного списка.
8. 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 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).

9. Поставьте флажок *Активный*.
10. Нажмите *Сохранить*.
11. На вкладке *Разрешения* назначьте права на список наблюдения, указав роли пользователей, которые смогут изменять/просматривать его настройки.



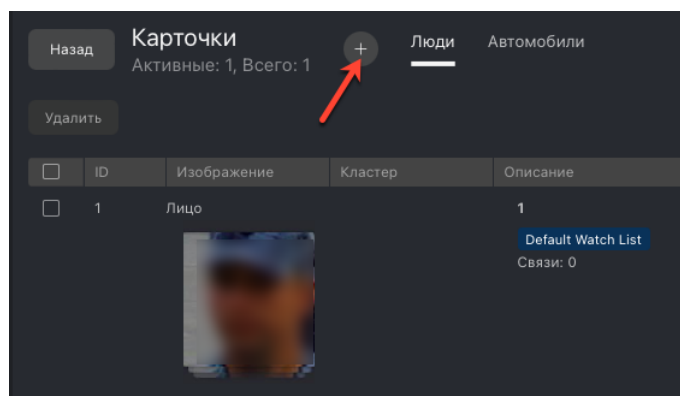
12. Нажмите *Сохранить*.

3.4.3 Создание карточки человека

Для того чтобы создать карточку человека вручную, выполните следующие действия:

Важно: Для того чтобы иметь возможность прикреплять изображения силуэта к карточке человека, необходимо включить функцию *детектирования силуэтов*.

1. В веб-интерфейсе перейдите на вкладку *Карточки*.
2. Нажмите *Люди*.
3. Нажмите *+*.



4. Добавьте изображения лица/силуэта человека. Поддерживаемые форматы: WEBP, JPG, BMP, PNG.

Важно: Лицо или силуэт на изображении должны быть надлежащего качества, т. е. в положении, близком к фронтальному. Расстояние между зрачками на изображении лица: 60 px. При несоответствии фотографии требованиям будет выведено сообщение с описанием ошибки.

The screenshot shows the 'Создать карточку' (Create Card) interface. At the top, there is a 'Назад' (Back) button and a title 'Создать карточку'. Below the title is a hint: 'Много карточек для загрузки? Попробуйте пакетную загрузку карточек'. On the right, there is a tab 'Основное' (Main). The form is divided into several sections: 'Фотографии' (Photos) with two photo thumbnails and a '+' icon; 'Лица' (Faces) with two face thumbnails and a '+' icon; 'Силуэты' (Silhouettes) with a '+' icon; 'Файлы' (Files) with a '+' icon; 'Имя' (Name) with a text field containing 'Лара Крофт'; 'Комментарий' (Comment) with a text area; 'Списки наблюдения' (Watch Lists) with a dropdown menu showing 'Default Watch List' and '+' and '-' icons; and a checkbox 'Активная' (Active) which is checked. At the bottom is a 'Сохранить' (Save) button.

5. Прикрепите имеющиеся отношение к делу файлы.
6. Введите имя карточки. При необходимости добавьте комментарий.
7. Из раскрывающегося списка *Списки наблюдения* выберите список, в который нужно добавить карточку (или несколько списков, по очереди).
8. Поставьте флажок *Активная*. Если карточка неактивна, она не будет использоваться для мониторинга.
9. Нажмите *Сохранить*.

Примечание: Если похожая карточка уже существует в картотеке, вам будут доступны следующие действия:

- добавление объектов из новой карточки в уже существующую и отмена создания новой карточки
- создание новой карточки

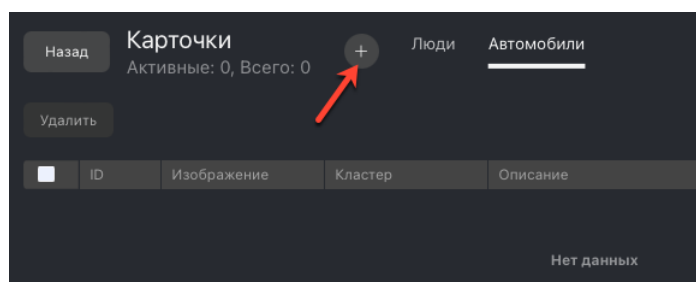
- отмена создания карточки

3.4.4 Создание карточки автомобиля

Важно: Для того чтобы карточки автомобилей были доступны для создания, необходимо включить функцию *детектирования автомобилей*.

Для того чтобы создать карточку автомобиля вручную, выполните следующие действия:

1. В веб-интерфейсе перейдите на вкладку *Карточки*.
2. Нажмите *Автомобили*.
3. Нажмите *+*.



4. Добавьте изображения автомобиля. Поддерживаемые форматы: WEBP, JPG, BMP, PNG.

Назад **Создать карточку** Основное

Много карточек для загрузки? Попробуйте пакетную загрузку карточек

Фотографии Автомобили

Файлы

Имя
Бабушкина машина

Комментарий

Регистрационный номер
XXX452A

Списки наблюдения
Списки наблюдения

☒ Активная

Сохранить

5. Прикрепите имеющиеся отношение к делу файлы.
6. Введите имя карточки. При необходимости добавьте комментарий.
7. При необходимости укажите номер автомобиля.
8. Из раскрывающегося списка *Списки наблюдения* выберите список, в который нужно добавить карточку (или несколько списков, по очереди).
9. Поставьте флажок *Активная*. Если карточка неактивна, она не будет использоваться для мониторинга.
10. Нажмите *Сохранить*.

Примечание: Если похожая карточка уже существует в картотеке, вам будут доступны следующие действия:

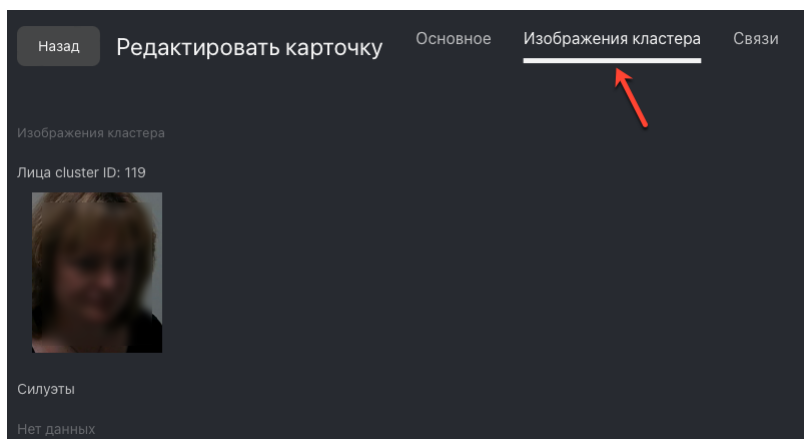
- добавление объектов из новой карточки в уже существующую и отмена создания новой карточки
 - создание новой карточки
 - отмена создания карточки
-

3.4.5 Просмотр сопоставленных кластеров

Важно: Для того чтобы иметь возможность просматривать сопоставленные кластеры, необходимо *включить кластеризацию* соответствующих типов объектов.

В процессе работы система может дополнять карточки совпадающими с ними *кластерами*.

Для просмотра кластера, сопоставленного с человеком или автомобилем, откройте нужную карточку и перейдите в ней на вкладку *Изображения кластера*.



В карточке человека вы увидите сопоставленный кластер лиц и кластер силуэтов. Сопоставленный кластер изображений автомобиля появится в карточке автомобиля.

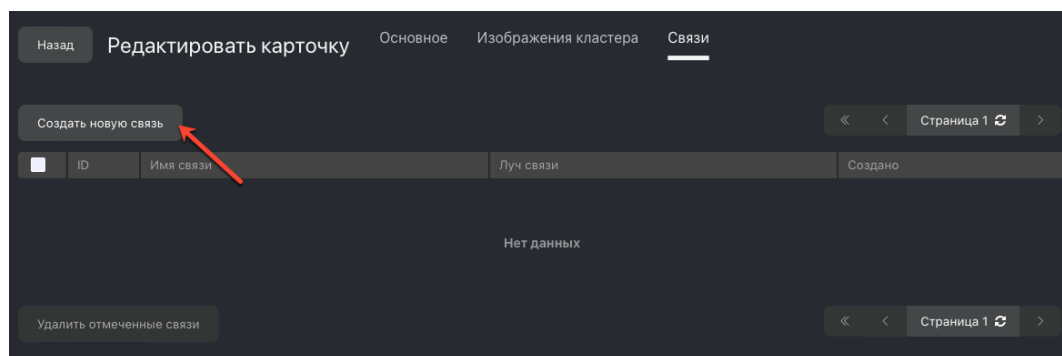
3.4.6 Связи карточки

Вы можете установить связи между карточками путем создания общей категории, к которой они относятся. Например, категория «Домохозяйство» может включать в себя всех членов семьи и их транспортные средства.

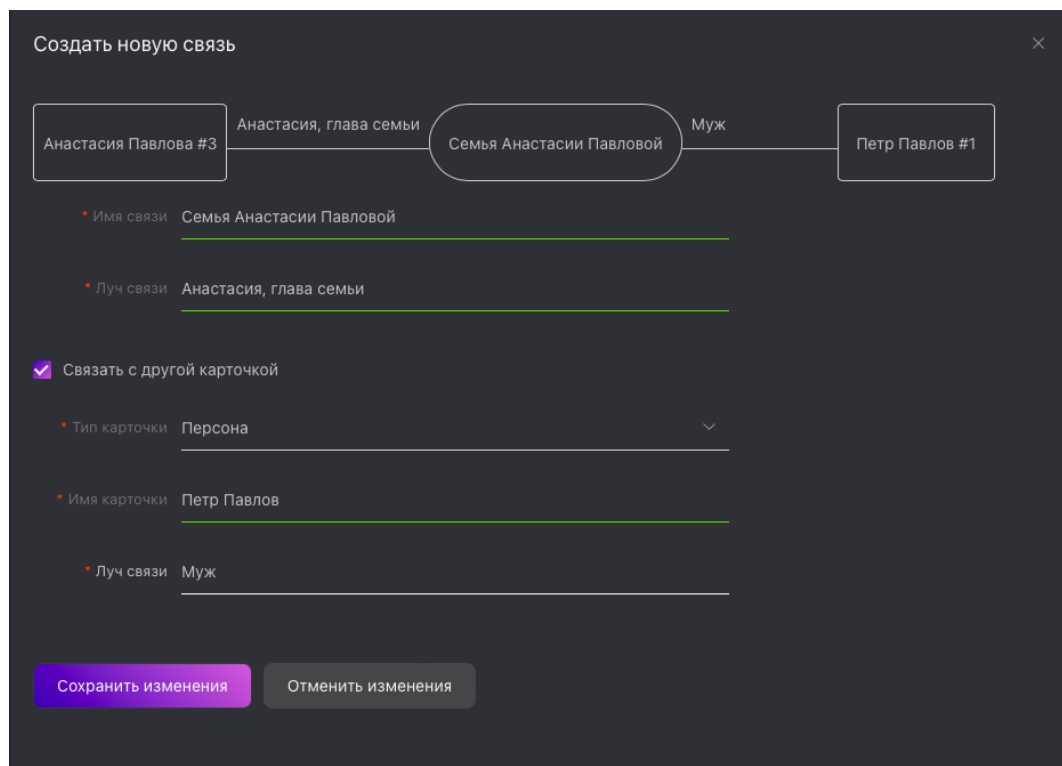
Карточку можно отнести к любому количеству категорий.

Для того чтобы установить связь карточки с категорией, выполните следующие действия:

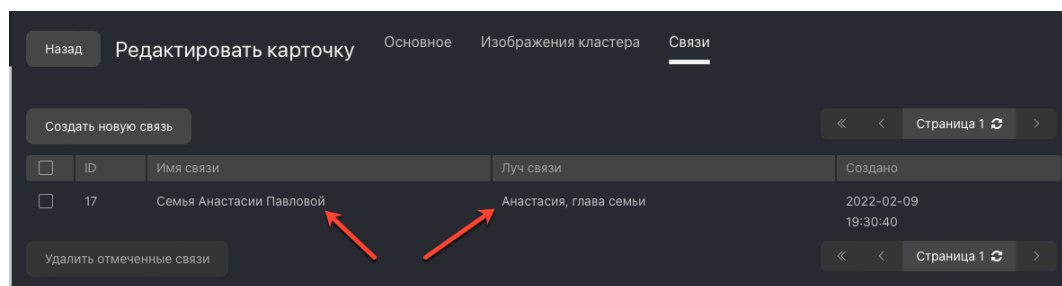
1. Откройте карточку для редактирования, щелкнув по ней в списке. Перейдите на вкладку *Связи*.
2. Нажмите *Создать новую связь*.



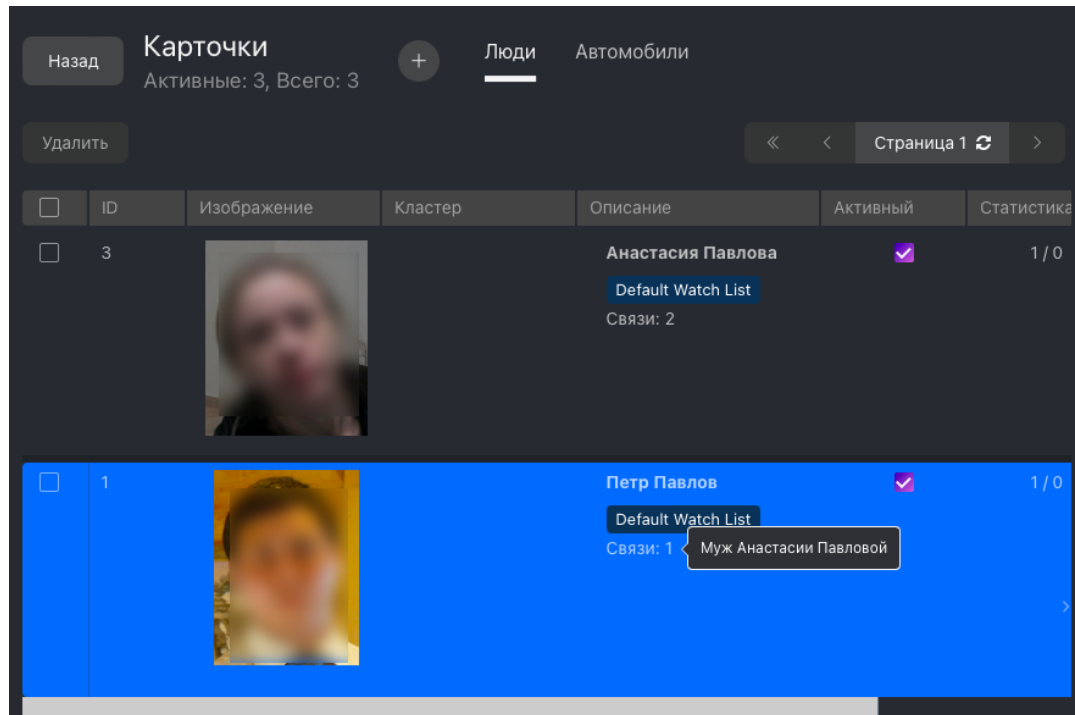
3. В поле *Имя связи* введите название категории. Если в системе уже существуют соответствующие категории, вы сможете увидеть их и выбрать одну из них после того, как начнете вводить название.



4. В поле *Луч связи* опишите связь между карточкой и категорией.
5. Если есть другая карточка, относящаяся к той же категории, вы можете оперативно добавить ее в этом же окне. Выполните следующие действия:
1. Поставьте флажок *Связать с другой карточкой*.
 2. Выберите тип карточки: *Человек* или *Автомобиль*.
 3. Начните вводить имя карточки в одноименном поле. В результате система автоматически предложит карточки с соответствующими именами. Выберите одну из них.
 4. В поле *Луч связи* опишите связь между карточкой и категорией.
6. Нажмите *Сохранить*. Созданные связи отобразятся на вкладке *Связи* и в списке карточек.



ID	Имя связи	Луч связи	Создано
17	Семья Анастасии Павловой	Анастасия, глава семьи	2022-02-09 19:30:40
		Муж	



Примечание: При наличии нескольких карточек, относящихся к одной категории, повторите данную пошаговую инструкцию в отношении их всех.

3.4.7 Пакетная загрузка карточек

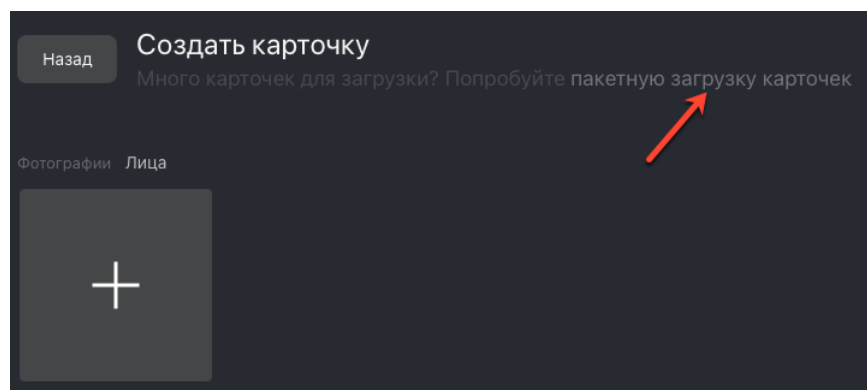
Для создания большого количества карточек используйте функционал пакетной загрузки. Выполните следующие действия:

Совет: Если количество загружаемых фотографий превышает 10000, используйте *пакетную загрузку через консоль*. В текущей версии такая загрузка доступна только для лиц.

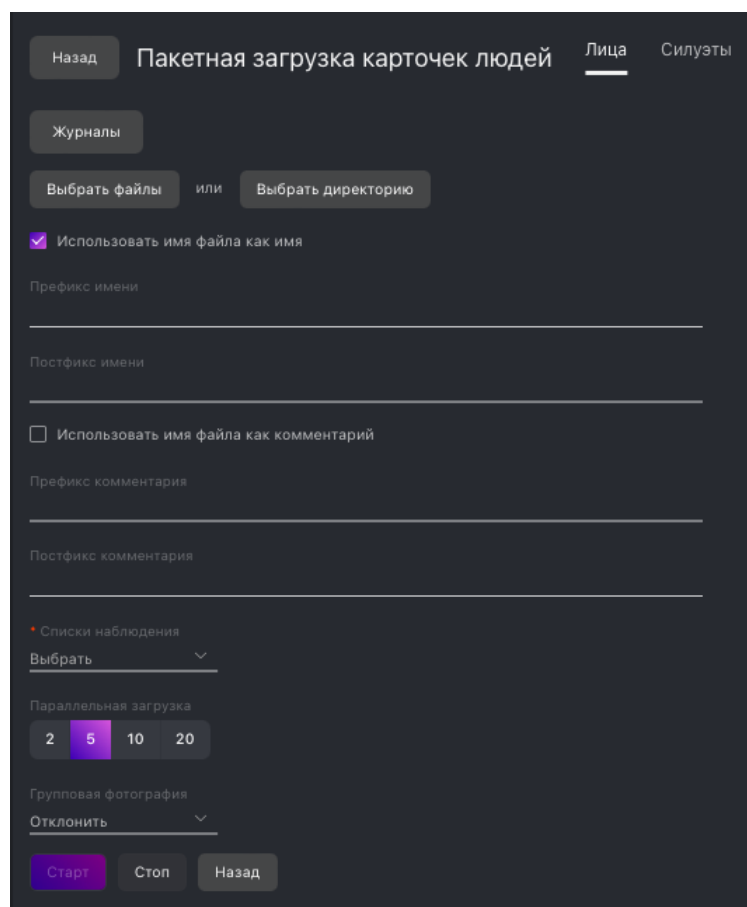
Важно: Поддерживаемые форматы: WEBP, JPG, BMP, PNG.

Важно: Лицо или силуэт на изображении должны быть надлежащего качества, т. е. в положении, близком к фронтальному. Расстояние между зрачками на изображениях лиц: 60 px. При несоответствии фотографии требованиям будет выведено сообщение с описанием ошибки.

1. В веб-интерфейсе перейдите на вкладку *Карточки*.
2. Нажмите *Люди* или *Автомобили*, в зависимости от типа карточек, которые вы добавляете.
3. Нажмите *+*.
4. Нажмите *Попробуйте пакетную загрузку карточек*.



5. Для карточек людей укажите подлежащие детектированию объекты: лица или силуэты.
6. Выберите фотографии для загрузки пофайлово или укажите каталог с фотографиями.



7. Имена файлов с фотографиями можно использовать как основу для имен и/или комментариев в создаваемых карточках. Выберите нужный вариант(ы). Затем настройте правило формирования имени и/или комментария, добавив пользовательский префикс и/или постфикс к имени файла.

Совет: Во избежание слияние 3-х слов в одно, используйте символ подчеркивания или пробел

в префиксе и постфиксе.

8. Из раскрывающегося списка *Списки наблюдения* выберите список (или несколько списков, по очереди), в который следует добавить создаваемые карточки.
9. В параметре *Параллельная загрузка* задайте количество потоков загрузки фотографий. Чем больше потоков, тем быстрее будет завершена загрузка, однако также потребуются и большее количество ресурсов.
10. Из раскрывающегося списка *Групповая фотография* выберите, как должна поступить система при наличии нескольких объектов на фотографии: отклонить фотографию, загрузить самый большой объект, загрузить все объекты.
11. Для запуска пакетного создания карточек нажмите на кнопку *Старт*.

Важно: Для просмотра лога пакетной загрузки нажмите на кнопку *Журналы*. Затем при необходимости можно скачать лог в формате `.csv`.

Журналы пакетной загрузки

Назад Удалить << < Страница 1 > >>

<input type="checkbox"/>	Id	Имя	Создано	Количество успешных	Количество ошибок	Скачать csv
<input type="checkbox"/>	3	admin-1562050597638000331	2019-07-02 11:56:36	3	0	Скачать
<input type="checkbox"/>	2	admin-1562037158774000980	2019-07-02 08:12:38	120	2	Скачать
<input type="checkbox"/>	1	admin-1558353930674000104	2019-05-20 17:05:30	97	3	Скачать

<< < Страница 1 > >>

3.4.8 Фильтры карточек

Все созданные в FindFace Multi карточки отображаются на вкладке *Карточки*. Используйте следующие фильтры, чтобы отфильтровать их:

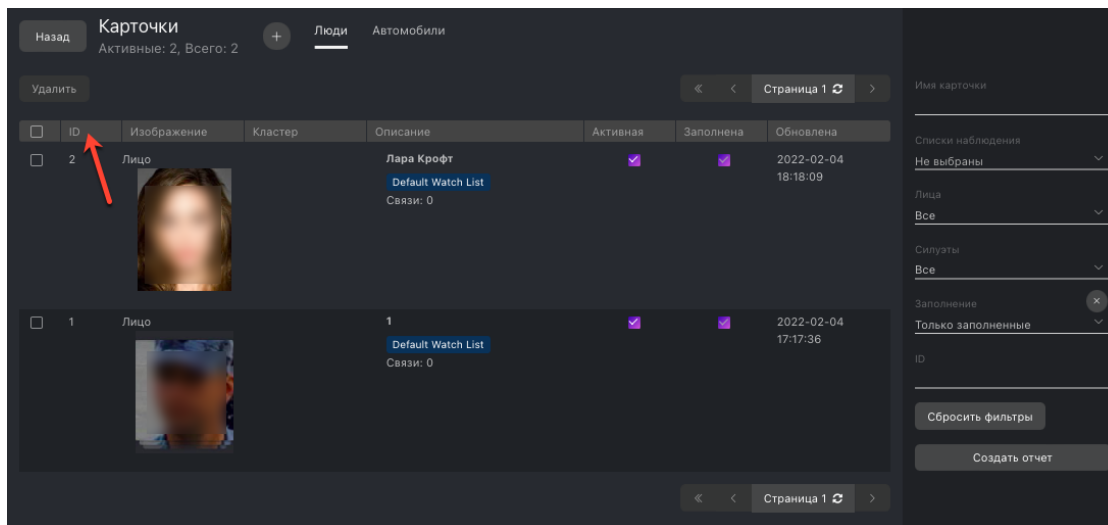
На вкладке *Люди* (карточки людей):

- *Имя карточки*: фильтровать карточки по имени.
- *Списки наблюдения*: отобразить карточки, включенные в выбранные списки наблюдения.
- *Лица*: фильтровать карточки по наличию изображения лица во вложении.
- *Силуэты*: фильтровать карточки по наличию изображения силуэта во вложении.
- *Заполнение*: отобразить карточки в зависимости от того, являются ли они автоматически сгенерированными (незаполненными) или заполненными. Подробнее см. раздел *Автокарточки и заполненные карточки*.
- *ID*: отобразить карточку с определенным ID.

На вкладке *Автомобили* (автомобильные карточки):

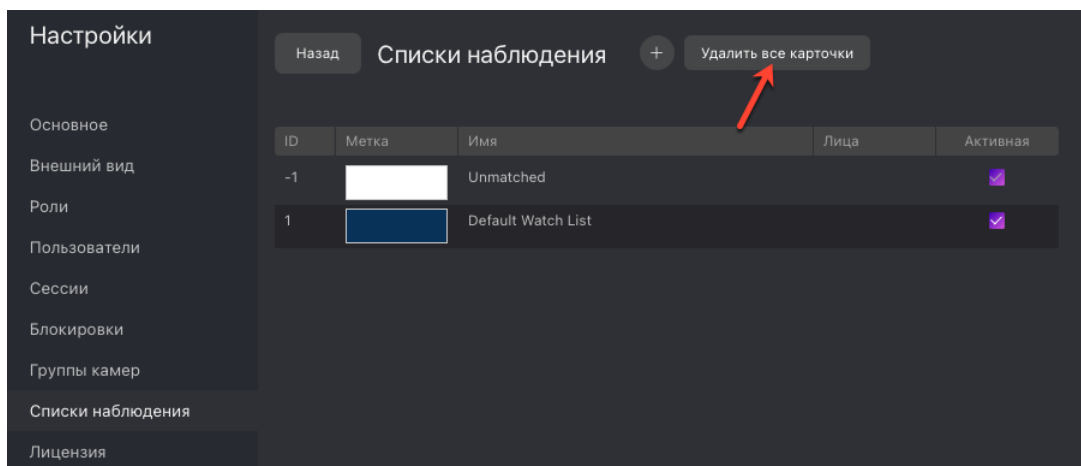
- *Имя карточки*: фильтровать карточки по имени.
- *Регистрационный номер*: фильтровать карточки по регистрационному номеру автомобиля.
- *Списки наблюдения*: отобразить карточки, включенные в выбранные списки наблюдения.
- *Автомобили*: фильтровать карточки по наличию изображения автомобиля во вложении.
- *Заполнение*: отобразить карточки в зависимости от того, являются ли они автоматически сгенерированными (незаполненными) или заполненными.
- *ID*: отобразить карточку с определенным ID.

Вы можете отсортировать карточки в списке по *ID*.



3.4.9 Очистка картотеки

Вы можете очистить картотеку в один клик. Для этого перейдите на вкладку *Настройки*. Выберите *Списки наблюдения*. Нажмите *Удалить все карточки*.



3.4.10 Отключение создания событий для определенных объектов

Иногда при мониторинге необходимо пропускать определенные объекты. Один из самых распространенных случаев — это объекты на рекламных носителях, расположенные в поле зрения камеры. Постоянно обнаруживаемые системой, они могут с легкостью переполнить ленту событий и базу данных.

Для того чтобы этого не произошло, выполните следующие действия:

1. *Создайте список наблюдения* для хранения объектов, не подлежащих детектированию. В его настройках установите флажок *Не создавать событий*.

Создать список наблюдения

Информация

Разрешения

Метка

Имя

Исключенные объекты

Группы камер

Не выбраны

Если список групп камер пустой, то список досье работает по всем камерам.

Комментарий

Порог срабатывания

☐ Порог срабатывания

Изменение данного параметра повлияет на работоспособность системы. Не трогайте, если вы не знаете точно, что делаете

☐ Требовать подтверждение события

☐ Включить звуковое оповещение

☒ Не создавать события

☒ Активный

Сохранить

Назад

2. Для каждого такого объекта создайте карточку и добавьте ее в созданный список наблюдения.

3.4.11 Автокарточки и заполненные карточки

В зависимости от способа создания выделяют автоматически сгенерированные (автокарточки) и заполненные карточки.

Автокарточки

Автокарточки создаются системой автоматически для кластеров, не имеющих совпадений в картотеке.

Примечание: По умолчанию создание автокарточек отключено. Чтобы включить его, установите 'CREATE_AUTO_CARDS': True в файле конфигурации /etc/findface-security/config.py и перезапустите findface-security.

```
sudo vi /etc/findface-security/config.py

# create auto cards for unmatched clusters
'CREATE_AUTO_CARDS': True,
```

```
sudo systemctl restart findface-security.service
```

На вкладке *Основное* в автокарточках нет загруженных изображений, поэтому такие карточки также называют незаполненными.

К незаполненным карточкам не применяются следующие фильтры:

- *Лица* -> *Только с лицами*
- *Силуэты* -> *Только с силуэтами*
- *Автомобили* -> *Только с автомобилями*

Заполненные карточки

В противоположность автокарточкам, в заполненных карточках есть загруженные изображения лица и силуэта человека или изображение автомобиля.

Для того чтобы разделить автокарточки и заполненные карточки в картотеке, используйте фильтр *Заполнение*.

Для того чтобы сделать автокарточку заполненной, загрузите в нее изображение того же объекта и измените имя карточки.

3.5 Счетчики лиц, силуэтов и машин. Определение дистанции

Важно: Для подсчета человеческих силуэтов или автомобилей нужно предварительно включить детектирование *силуэтов* или *машин*.

FindFace Multi позволяет подсчитывать лица, силуэты и автомобили 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 машины на статических скриншотах, которые делаются с заданным интервалом. Счетчик показывает, как количество лиц и силуэтов меняется со временем.

Вы можете вести подсчет лиц/силуэтов/автомобилей на каждой отдельной камере или же совместно на всех выбранных камерах.

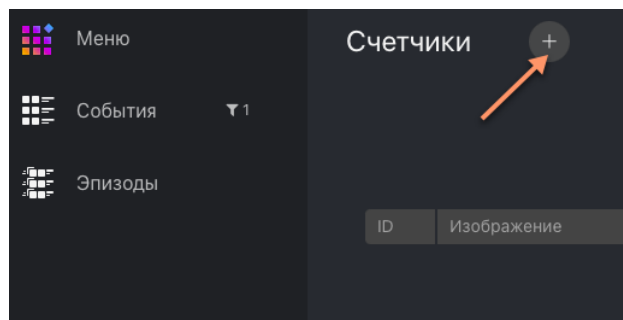
В этом разделе:

- *Создание счетчика*
- *Калибровка счетчика для определения дистанции*
- *График счетчика*
- *Мониторинг работы счетчиков*
- *Работа с записями счетчика*
- *Настройка вебхука для счетчика*
- *Настройка счетчиков*

3.5.1 Создание счетчика

Для создания счетчика выполните следующие действия:

1. Перейдите на вкладку *Счетчики*.
2. Нажмите **+**.



3. Укажите имя счетчика.
4. Выберите одну или несколько групп камер для подсчета.

Совет: По умолчанию подсчет будет проводиться по всем камерам в указанных группах. При необходимости вы можете отключить подсчет по определенным камерам.

5. Задайте интервал в секундах между двумя последовательными скриншотами, используемыми для подсчета лиц и силуэтов.
6. Установите флажок *Определение лиц* для подсчета лиц.
7. Установите флажок *Определение силуэтов* для подсчета силуэтов. Детектирование силуэтов должно быть *включено*.

8. Установите флажок *Определение автомобилей* для подсчета автомобилей. Детектирование автомобилей должно быть *включено*.
9. Установите флажок *Определение дистанции* для измерения расстояния между силуэтами. Данный флажок доступен в том случае, если установлен флажок *Определение силуэтов*.
10. Убедитесь, что для счетчика установлен флажок *Активный*.
11. Нажмите *Сохранить*.

Создать счетчик

Информация

Области интереса

График

Имя

Офис

Группы камер

NTL office

Камеры

NTL elevator door

NTL 9 floor entrance

NTL office entrance

NTL exit

Openspace

Интервал подсчета (секунды)

5

Определение лиц

Определение силуэтов

Чтобы использовать счётчик по силуэтам необходимо включить детектирование силуэтов. Для этого ознакомьтесь с документацией (раздел "Включение детектирования силуэтов"), или обратитесь к администратору вашей установки FindFace Multi.

Определение автомобилей

Чтобы использовать счётчик по автомобилям необходимо включить детектирование автомобилей. Для этого ознакомьтесь с документацией (раздел "Распознавание автомобилей и их атрибутов"), или обратитесь к администратору вашей установки FindFace Multi.

Активный

Если для камеры включены счетчики, они активны всегда.

Определение дистанции

После включения функции "Определение дистанции" необходимо нажать кнопку "Сохранить". После сохранения счетчика нужно произвести настройку счетчика: ввести время калибровки и нажать кнопку "Откалибровать счетчик".

Статус

Ожидание создания модели

60

Откалибровать счетчик

Задать продолжительность калибровки (секунды).

Минимальное время - 15, максимальное - 300.

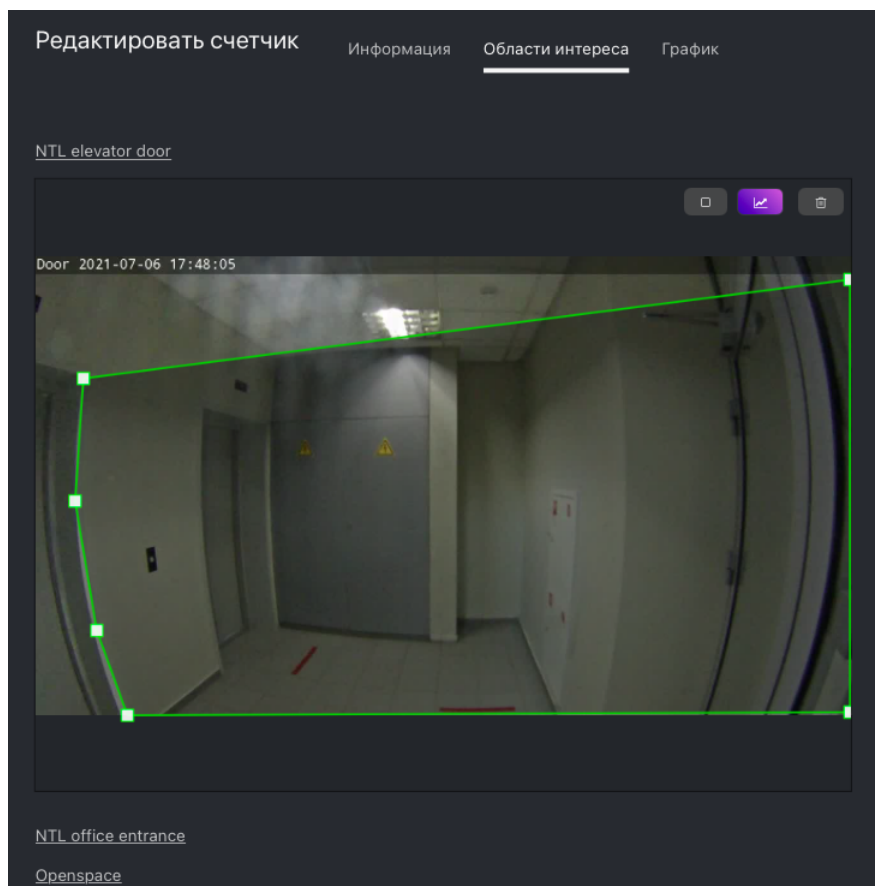
Удалить калибровку

Сохранить

Назад

Удалить

12. (Опционально) Перейдите на вкладку *Области интереса*, чтобы задать регион отслеживания лиц и силуэтов в поле зрения камер(ы).



3.5.2 Калибровка счетчика для определения дистанции

Если включено *Определение дистанции*, необходимо выполнить калибровку счетчика.

Для калибровки счетчика:

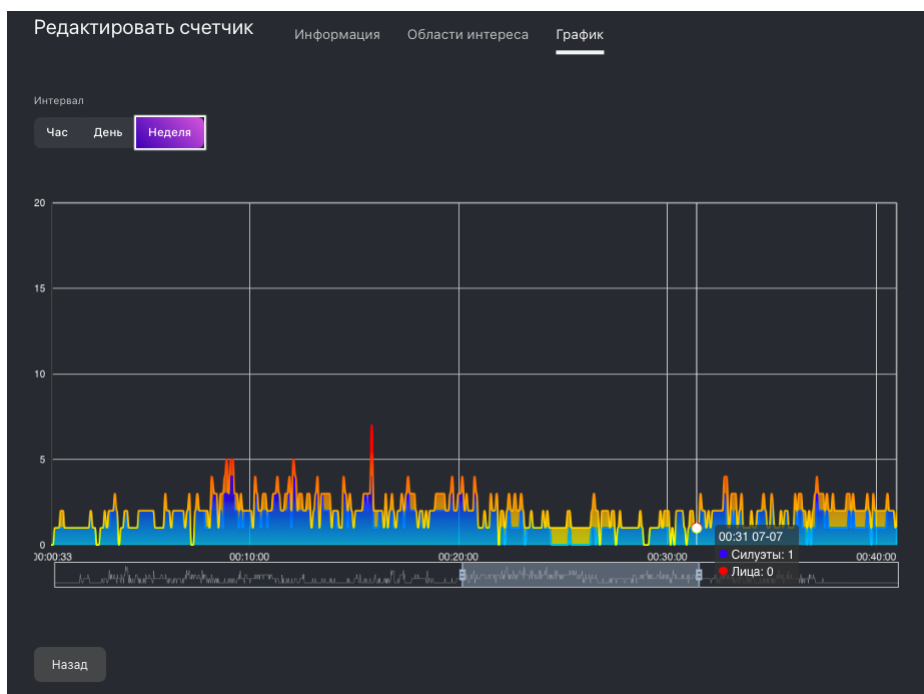
1. Попросите человека встать перед выбранной камерой в полный рост.
2. Попросите пройти по зоне, фиксируемой камерой, по которой в дальнейшем будут рассчитываться расстояния. Для достижения наилучшего качества человек должен ходить по ровной поверхности.
3. Введите период калибровки счетчика (от 15 до 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.

3.5.3 График счетчика

Для того чтобы увидеть график счетчика за последние час, сутки или неделю, перейдите на вкладку *График* в настройках счетчика.



3.5.4 Мониторинг работы счетчиков

Мониторинг работы счетчиков выполняется на вкладке *Счетчики*.

ID	Изображение	Тип счётчика	Лица	Силуэты	Дистанция	Статус	Имя	Активен
3	-	По одной камере	0	0	- / - / -	0 / 0 Калиброван	Entrance counter	<input type="checkbox"/>
2		По одной камере	1	0	- / - / -	0 / 5 Ожидание синхронизации	Dev space, NTL office counter	<input checked="" type="checkbox"/>
1		По нескольким камерам	0	1	- / - / -	0 / 10 Не удалось откалибровать	Дистанция	<input checked="" type="checkbox"/>

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 `/etc/findface-security/config.py`:

```
sudo vi /etc/findface-security/config.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,
    ...
}
```

3.5.5 Работа с записями счетчика

Статические скриншоты, которые делаются счетчиком, с количеством обнаруженных лиц, силуэтов и машин, 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*. В соответствующей счетчику строке нажмите на любой столбец, кроме ID (ведет в настройки счетчика).

Назад

Записи счетчика

«

<

Страница 1

>

»

ID	Изображение	Лица	Силуэты	Автомобили	Дистанция	Дата
6251		0	0	0	-/-/-	2021-06-21 18:16:51
625С		0	0	0	-/-/-	2021-06-21 18:16:51
624Е		0	0	0	-/-/-	2021-06-21 18:16:51
623Е		0	0	0	-/-/-	2021-06-21 18:16:4Е
623Е		0	0	0	-/-/-	2021-06-21 18:16:4Е

Счетчик

Офис

Камеры

Не выбраны

Старт

Конец

Количество лиц

От

Количество силуэтов

От

Количество автомобилей

От

ID записей счетчика

Минимальная дистанция

От

До

Максимальная дистанция

От

До

Средняя дистанция

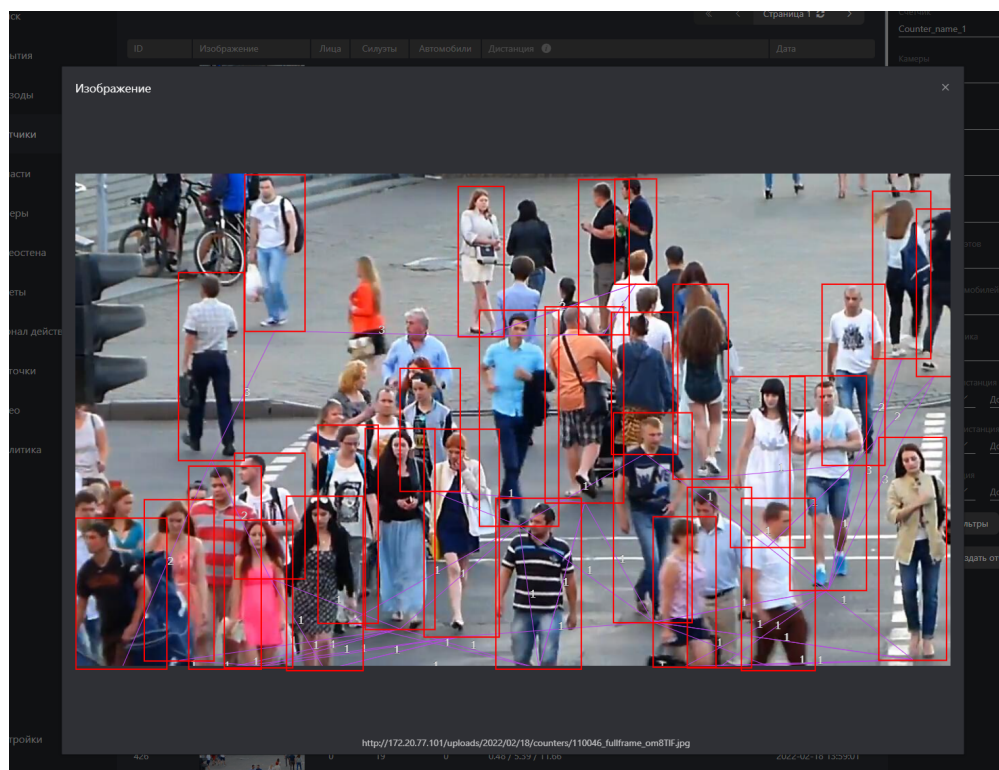
От

До

Сбросить фильтры

Создать отчет

Вы можете увеличить скриншот, нажав на него. Если включено определение дистанции, на скриншоте будут зафиксированы прямоугольники вокруг силуэтов и расстояние между ними.



Для работы с записями счетчика используйте следующие фильтры:

- Счетчик
- Камеры

- Группы камер
- Период времени
- Количество лиц в записи
- Количество силуэтов в записи
- Количество автомобилей в записи
- ID записи
- Минимальная дистанция
- Максимальная дистанция
- Средняя дистанция

3.5.6 Настройка вебхука для счетчика

Для того чтобы автоматизировать процесс сбора статистики по счетчику, *настройте вебхук* на срабатывание при определенном количестве лиц, силуэтов и автомобилей в записи счетчика.

См.также:

- *Распознавание силуэтов и их атрибутов*
- *Распознавание автомобилей и их атрибутов*
- *Вебхуки*

3.5.7 Настройка счетчиков

Для настройки счетчиков откройте файл конфигурации `/etc/findface-security/config.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 автомобили, **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 /etc/findface-security/config.py

# deduplicate objects on multi-camera counters
'COUNTERS_DEDUP_OPTIONS': {
    'face': {
        'enabled': False,
        'threshold': 0.723, # model: [kiwi_320]
```

(continues on next page)

(продолжение с предыдущей страницы)

```

    },
    'body': {
        'enabled': False,
        'threshold': 0.65, # model: [andariel]
    },
    'car': {
        'enabled': False,
        'threshold': 0.65, # model: [alonso]
    },
},
# 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 сервис `findface-security`.

```
sudo systemctl restart findface-security.service
```

3.6 Управление областями

Важно: Для того чтобы использовать функциональность областей, предварительно включите *обна-
ружение симуэтов*.

FindFace Multi позволяет отслеживать присутствие людей на камерах в определенных областях по заданным правилам и расписаниям. Как только система обнаруживает, что ситуация в области соответствует правилу, она создает так называемую активацию области.

Области — отличный инструмент для решения таких задач, как предотвращение длинных очередей в торговых точках, предотвращение краж на предприятиях в нерабочее время, контроль опасных зон, учет рабочего времени и многих других.

В этом разделе:

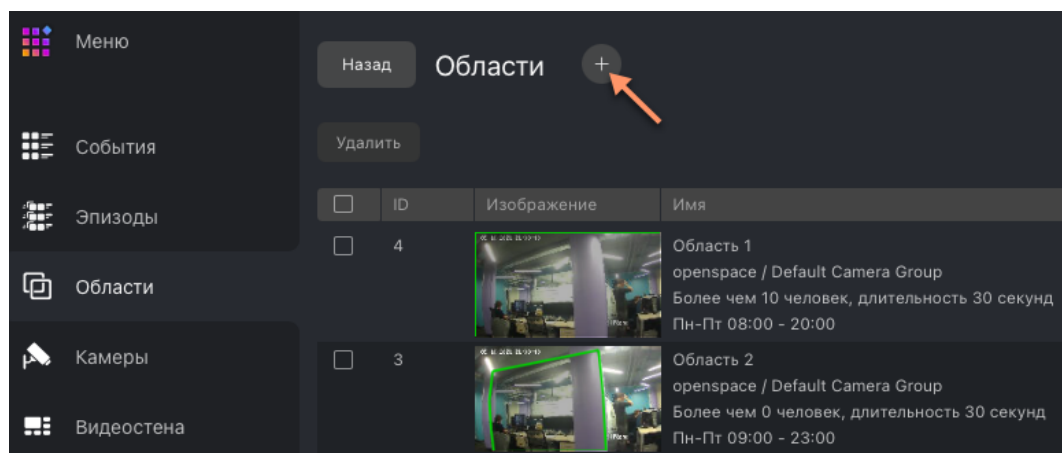
- *Создание области*
- *Редактирование области*
- *Работа с активациями области*
- *Отчеты о рабочем времени по областям*
- *Настройка вебхука для области*

- *Настройка областей*

3.6.1 Создание области

Для создания области выполните следующие действия:

1. Перейдите на вкладку *Области*.
2. Нажмите *+*.



3. Укажите название области.

Назад Редактирование области

* Имя
Область 1

* Камеры
Entrance

Области интереса
Редактировать камеру "Entrance"

Условие активации
* Тип
Более чем

* Количество человек
0

* Длительность (секунды)
30

Расписание
+ -

* Дни недели
Сб Вс

* С
⌚ 07:15
Время должно указываться в формате UTC

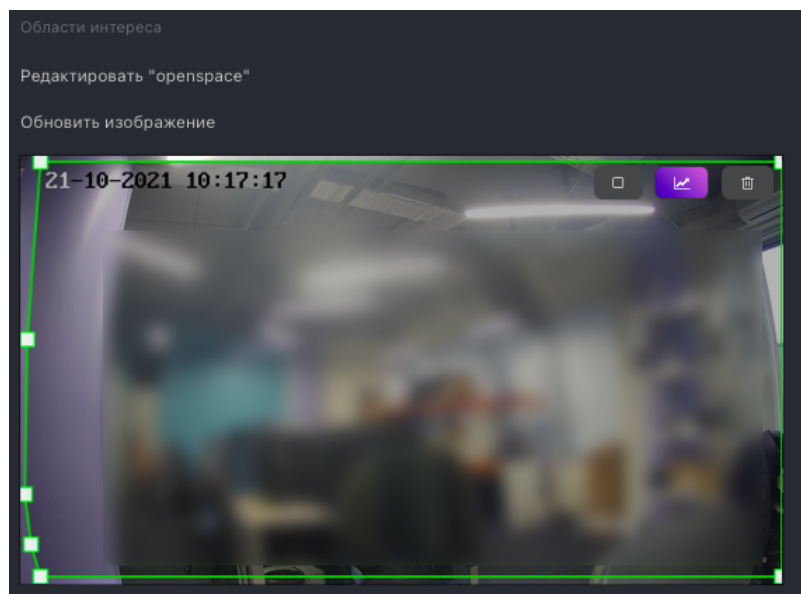
* До
⌚ 21:45
Время должно указываться в формате UTC

Сохранить




4. Выберите одну или несколько камер для мониторинга области.

Совет: Вы можете создать несколько областей на одной и той же камере для мониторинга физической области по разным правилам и расписаниям.

5. Укажите область интереса в поле зрения камер(ы).



Примечание: Используйте следующие кнопки:

-  : начертить прямоугольник
 -  : начертить многоугольник по точкам
 -  : удалить созданную область
-

6. Настройте условие активации области:

- *Менее чем:* область будет активирована, если количество человек в течение определенного времени будет меньше порогового.
- *Более чем:* область будет активирована, если количество человек в течение определенного времени будет больше порогового.
- *Количество человек:* пороговое количество человек.
- *Длительность:* минимальный период времени в секундах, в течение которого должно выполняться условие срабатывания, чтобы область активировалась. В течение этого периода количество человек должно быть больше/меньше порогового значения.

Примечание: Например, условие *Более чем 5* *Продолжительность 30*, примененное к кассе, приводит к активации области, если более пяти покупателей более 30 секунд стоят в очереди.

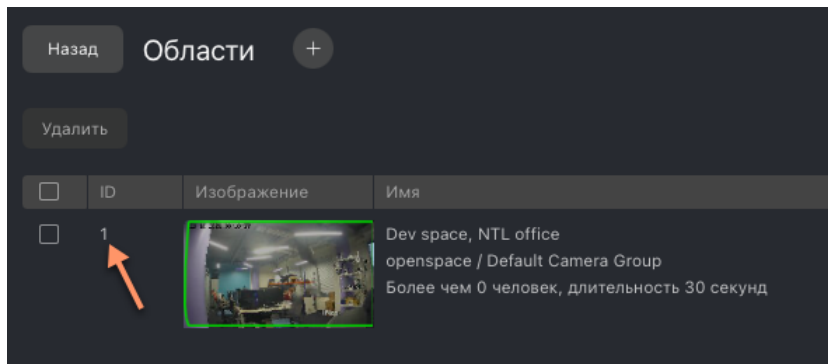
- *Расписание:* создайте одно или несколько расписаний мониторинга. Укажите дни недели и время, в течение которого область будет под наблюдением. Если расписание не задано, наблюдение будет постоянным.

Примечание: Если ситуация в области продолжает соответствовать условию активации после планового периода мониторинга, область будет оставаться активной, пока данная ситуация не завершится.

7. Нажмите *Сохранить*.

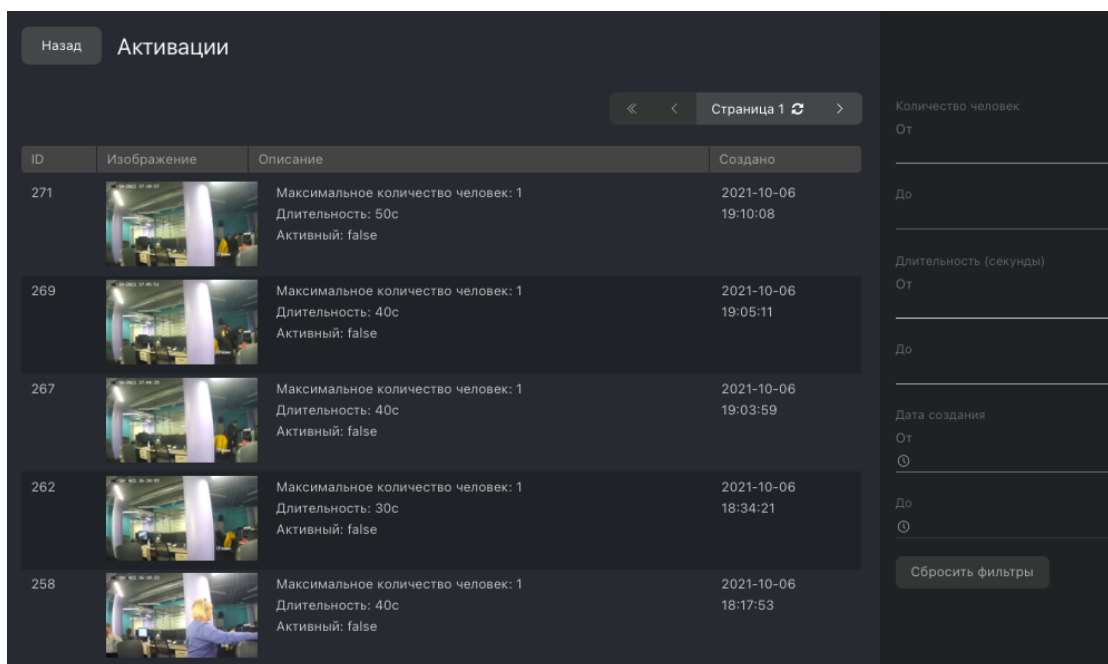
3.6.2 Редактирование области

Чтобы открыть настройки области, нажмите на столбец *ID* в списке областей.



3.6.3 Работа с активациями области

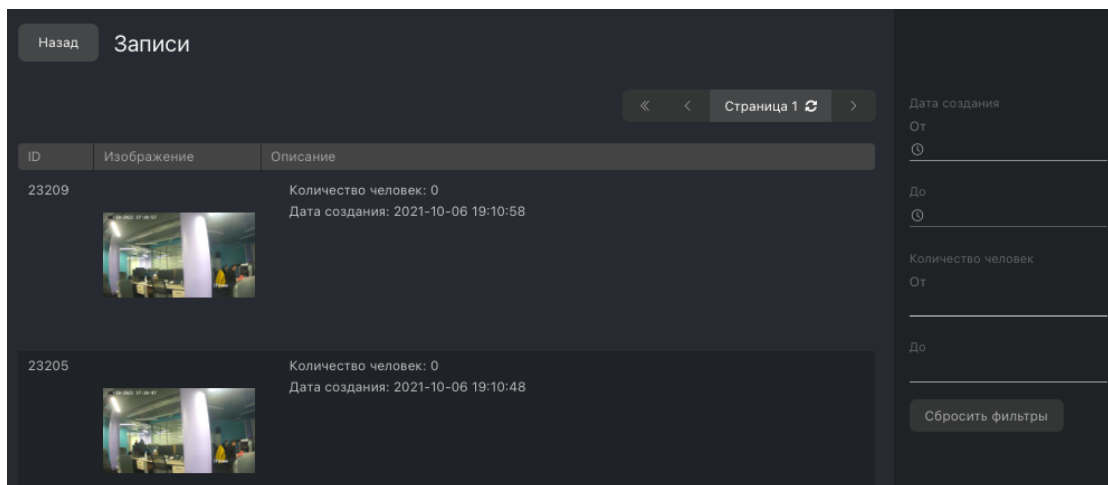
Как только ситуация в области начнет соответствовать условию срабатывания, система активирует область. Для просмотра списка активаций области перейдите на вкладку *Области*. В соответствующей области строке нажмите на любой столбец, кроме *ID* (ведет в настройки области).



При необходимости примените к списку активаций следующие фильтры:

- Количество обнаруженных в области людей (от/до)
- Продолжительность срабатывания (от/до)
- Дата и время создания активации (от/до)

Нажмите на активацию, чтобы просмотреть записи в ней. Записи представляют собой скриншоты с соответствующей камеры области, которые сохраняются вместе с данными о количестве людей и меткой времени.



Примечание: Записи сохраняются с 10-секундным интервалом на протяжении всего периода срабатывания.

Отфильтруйте список записей, используя следующие критерии:

- Количество людей в записи (от/до)
- Метка времени записи (от/до)

3.6.4 Отчеты о рабочем времени по областям

См. *Отчеты*.

3.6.5 Настройка вебхука для области

Для того чтобы автоматизировать работу с областями, *настройте вебхуки*.

3.6.6 Настройка областей

Вы можете настроить параметры сохранения снимков экрана (полных кадров), отображаемых в записях активации областей, и их миниатюр. Также можно включить дедупликацию силуэтов в многокамерных областях, чтобы избежать ситуации, когда один и тот же человек считается на нескольких камерах одновременно. Данная ситуация может привести к чрезмерно завышенному результату подсчета и, как следствие, к ложной активации/отсутствию активации области.

Для этого откройте файл конфигурации `/etc/findface-security/config.py` и измените следующие параметры:

- **AREA_DEDUP_OPTIONS:** набор опций, которые позволяют избежать дублирования силуэтов в многокамерных областях. Доступны две опции: **enabled** - включает дедупликацию силуэтов, **threshold** - определяет порог схожести, при котором силуэты считаются дубликатами.

- AREAS_SAVE_FULLFRAME определяет сохранение полных кадров в записях активаций: `always` (всегда), `never` (никогда).
- AREAS_THUMBNAIL_JPEG_QUALITY: качество JPEG миниатюр.
- AREAS_FULLFRAME_JPEG_QUALITY: качество JPEG полных кадров.

```
sudo vi /etc/findface-security/config.py

...
# deduplicate objects on multi-camera areas
'AREA_DEDUP_OPTIONS': {
    'enabled': False,
    'threshold': 0.65, # model: [andariel]
},
# areas full frame saving options:
# `always` - save always
# `never` - never save full frames
'AREAS_SAVE_FULLFRAME': 'always',
'AREAS_THUMBNAIL_JPEG_QUALITY': 75,
'AREAS_FULLFRAME_JPEG_QUALITY': 75,
...
```

После внесения изменений обязательно перезапустите сервис `findface-security`.

```
sudo systemctl restart findface-security.service
```

3.7 События и эпизоды распознавания объектов

Результат работы системы по части идентификации объектов на видеоизображении в режиме реального времени отображается на вкладках *События* и *Эпизоды*. Помимо работы с текущими событиями идентификации, данные вкладки также предоставляют доступ к истории событий.

Совет: Поиск объекта в списке событий и картотеке выполняется на вкладке *Поиск*.

Совет: Для идентификации объектов в архивных видео см. *Идентификация объектов в офлайн видео*.

3.7.1 Работа с событиями

Данный раздел посвящен вкладке *События*.

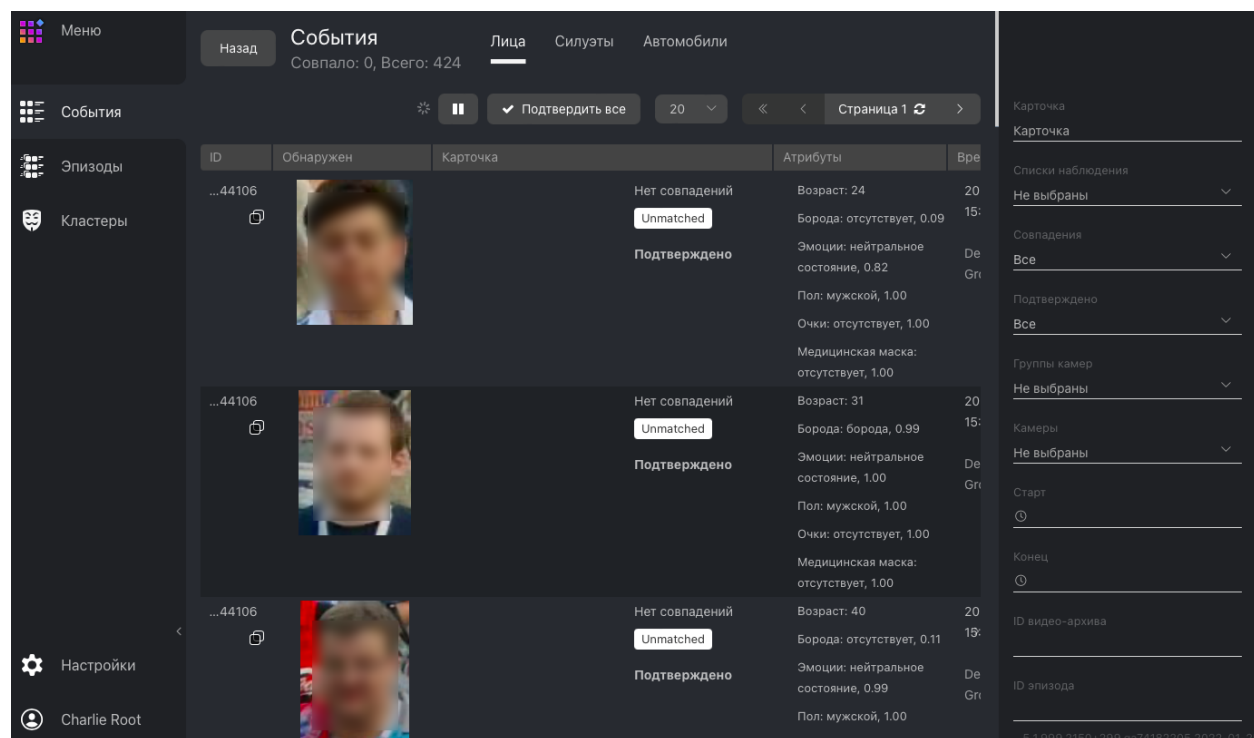
Важно: Вы можете *включить звуковые уведомления* для событий, связанных с конкретными списками наблюдения. В некоторых браузерах для того чтобы воспроизводился звук, вкладка с событиями должна оставаться в фокусе. Чтобы выделить вкладку, откройте ее и щелкните в любом месте страницы.

В этой главе:

- Просмотр событий
- Карточка события. Принятие события
- Карточка события. Поиск объекта
- Отчеты о рабочем времени по событиям
- Видеоплеер в уведомлении о событии


Просмотр событий

Как только объект будет обнаружен, вы увидите уведомление в одном из списков событий: *Лица*, *Силуэты* или *Автомобили*, в зависимости от типа объекта.



Уведомление может содержать разную информацию в зависимости от того, есть ли совпадение обнаруженного объекта с картотекой:

- Если совпадение не найдено: нормализованное изображение объекта, дата и время обнаружения объекта, группа камер.
- Если совпадение найдено: нормализованное изображение объекта, фотография из карточки, имя, степень схожести объектов, комментарий из карточки, список наблюдения, дата и время обнаружения объекта, группа камер.

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 активированного функционала распознавания.

- *Карточка*: отображать только события по определенной карточке.
- *Списки наблюдения*: отображать только события по определенному списку наблюдения.
- *Совпадения*: отображать только события с совпадением объектов/без совпадения объектов или все события.
- *Подтверждено*: отображать только принятые/непринятые или все события.
- *Камеры*: отображать только события по определенной камере.
- *Группы камер*: отображать только события по определенной группе камер.
- *Старт, Конец*: отображать только события, случившиеся в определенный период времени.
- *ID видеоархива*: отобразить события из видеоархива с определенным ID.
- *ID события*: отобразить событие с определенным ID.
- *ID эпизода*: отобразить события из эпизода с определенным ID.
- *Лимит*: отобразить заданное количество наиболее релевантных событий.

Специальные фильтры для лиц

- *Лучший кадр события*: отображать все события трека, только события с кадрами в реальном времени, только одно событие с лучшим кадром по окончанию трека.
- *Возраст*: отобразить события с людьми определенного возраста.
- *Пол*: отобразить события с людьми заданного пола.
- *Эмоции*: отобразить события с заданными эмоциями.
- *Очки*: фильтровать события по наличию очков на лице.
- *Борода*: фильтровать события по наличию бороды.
- *Liveness*: фильтровать события по liveness лица.
- *Медицинская маска*: фильтровать события по наличию медицинской маски.

Специальные фильтры для силуэтов

- *Головной убор*: отображать только события с людьми в головном уборе заданного типа: шапка/шляпа/кепка, капюшон/платок, без головного убора.
- *Верх одежды*: отображать только те события, в которых человек носит одежду верха заданной обобщенной категории: с длинными рукавами, с короткими рукавами, без рукавов.
- *Тип верха одежды*: отображать только те события, в которых человек носит одежду верха заданного типа: куртка, пальто, жилет, толстовка, футболка, рубашка, платье.
- *Цвет верха одежды*: отображать только события с людьми, одетыми в одежду заданного цвета (верх).

- *Низ одежды*: отображать только те события, в которых человек носит одежду низа заданного типа: брюки, юбка, шорты, неопределенный.
- *Цвет низа одежды*: отображать только события с людьми, одетыми в одежду заданного цвета (низ).

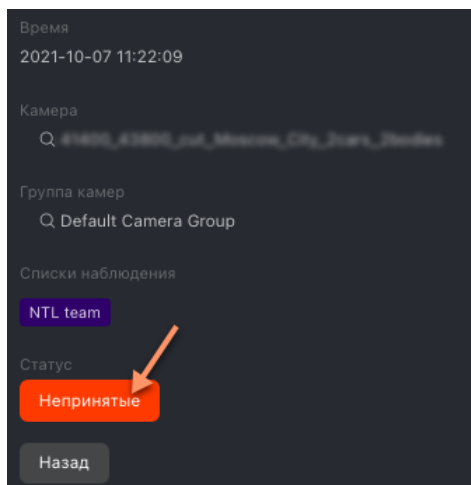
Специальные фильтры для автомобилей

- *Тип кузова*: отобразить события с автомобилями заданного типа кузова.
- *Цвет автомобиля*: отобразить события с автомобилями заданного цвета.
- *Страна*: отобразить события с автомобилями, зарегистрированными в заданной стране.
- *Регистрационный номер*: найти автомобиль с заданным номером.
- *Регион*: отобразить события с автомобилями, зарегистрированными в заданном регионе.
- *Марка*: фильтровать события по марке автомобиля.
- *Модель*: фильтровать события по модели автомобиля.
- *Спецтранспорт*: отображать только события с транспортными средствами, принадлежащими заданному типу: полиция, скорая помощь, спасательные службы, такси, прочие.

Карточка события. Принятие события

Для того чтобы перейти в карточку события из списка событий, щелкните в уведомлении по результату распознавания (*Нет совпадений* или имя из соответствующей карточки).

Карточка содержит ту же информацию, что и *уведомление*, а также предоставляет возможность принять событие. Для того чтобы это сделать, поставьте флажок *Подтверждение события*. Нажмите на кнопку *Сохранить*.



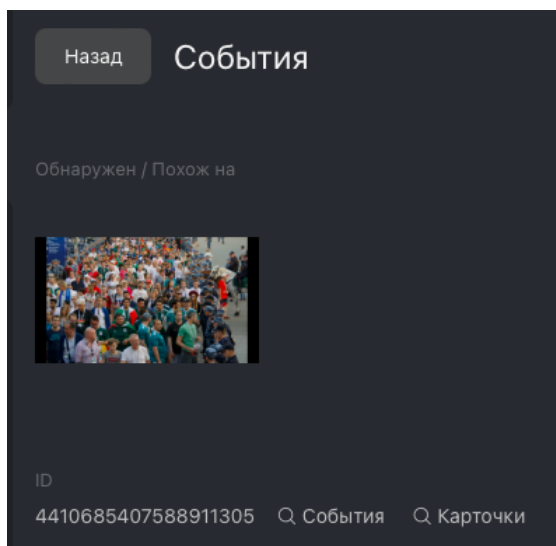
Совет: Если у обнаруженного объекта есть совпадения по картотеке, в соответствующую карточку можно перейти, щелкнув по имени в карточке события.

Совет: Для того чтобы принять все события, нажмите на кнопку  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 *События* или *Карточки*.



См.также:

- *Поиск объектов в базах данных*

Отчеты о рабочем времени по событиям

См. *Отчеты*.


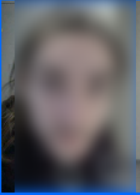
Видеоплеер в уведомлении о событии

Важно: Только при активированной функции *Видеомагнитофон*.


Если Видеомагнитофон *развернут* и *настроен* должным образом, при нажатии на миниатюру события появится видеоплеер, в котором начнется воспроизведение видео с данным событием.

Назад **События** Лица Силуэты Автомобили
Совпало: 0, Всего: 12710

☀️ ⏸️ ✓ Подтвердить все 20 << < Страница 1 > >

ID	Обнаружен	Карточка	Атрибуты	Вре
...44227		Нет совпадений Unmatched Подтверждено	Возраст: 29 Борода: борода, 1.00 Эмоции: нейтральное состояние, 0.98 Пол: мужской, 1.00 Очки: отсутствует, 0.98	20 19: De Gr
...44227		Нет совпадений Unmatched Подтверждено	Возраст: 29 Борода: борода, 1.00 Эмоции: нейтральное состояние, 0.99 Пол: мужской, 1.00 Очки: отсутствует, 0.98	20 19: De Gr

5fl-entrance



⏮️ ⏪️ ⏩️ ⏭️ ⏭️ ⏭️ Live

19:18 19:21 19:24 19:27 19:30 19:33 19:35:10 19:39 19:42 19:45 19:48

2022-03-18 🔍 🔍 Экспорт

Видеоплеер имеет интуитивно понятный дизайн. Подробнее см. *Просмотр видео с камеры в видеоплеере*.

3.7.2 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 автомобиля), 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*
- *Фильтрация событий по ID 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 автомобилями. Эпизоды с людьми содержат изображения лиц и силуэтов людей, обнаруженные в течение определенного периода времени, а эпизоды с автомобилями содержат изображения автомобилей.

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 автомобилями

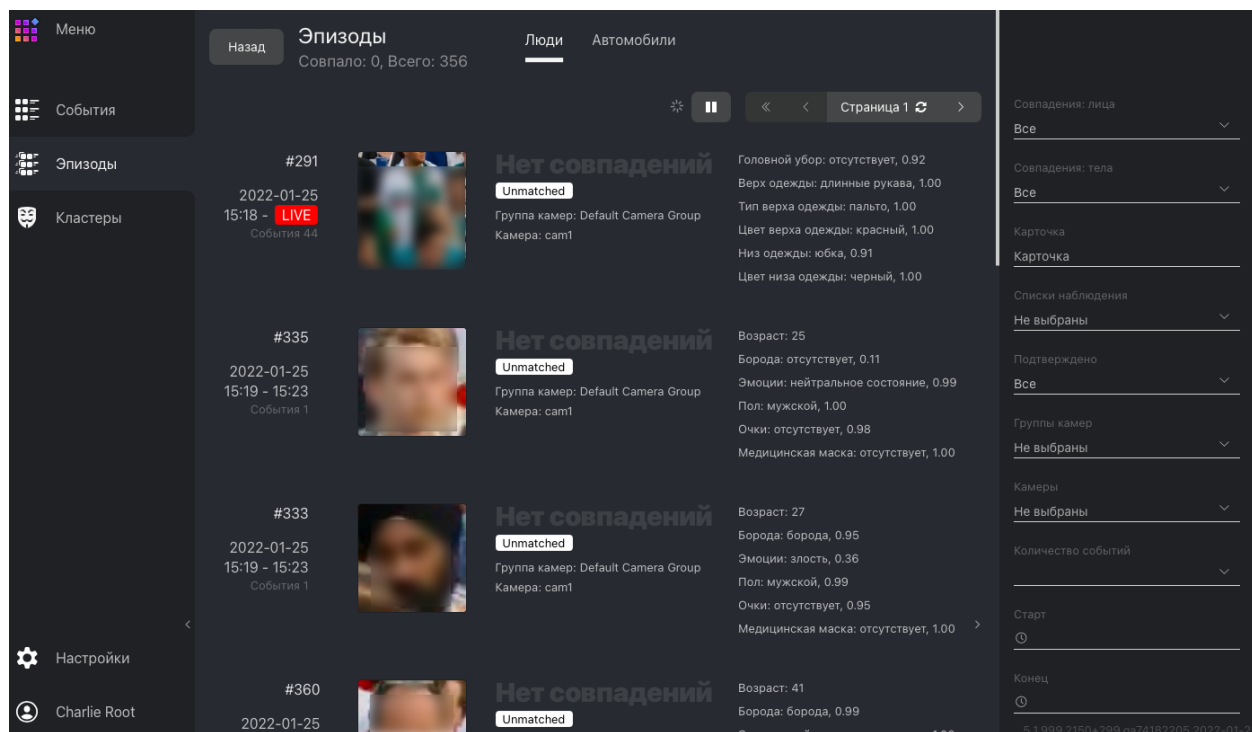
Совет: См. *Управление пользователями*.



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 автомобилями.

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 активированного функционала распознавания.

- *Карточка*: отобразить только эпизоды по определенной карточке.
- *Списки наблюдения*: отобразить только эпизоды по определенному списку наблюдения.
- *Совпадения*: отобразить только эпизоды с совпадением объектов/без совпадения объектов/все эпизоды.
- *Подтверждено*: отобразить только принятые/непринятые или все эпизоды.
- *Камеры*: отобразить только эпизоды по определенной камере.
- *Группы камер*: отобразить только эпизоды по определенной группе камер.
- *Старт*, *Конец*: отобразить только эпизоды, случившиеся в определенный период времени.
- *Количество событий*: отобразить эпизоды с заданным количеством событий.
- *ID видеоархива*: отобразить эпизоды из видеоархива с определенным ID.
- *ID эпизода*: отобразить эпизод с определенным ID.

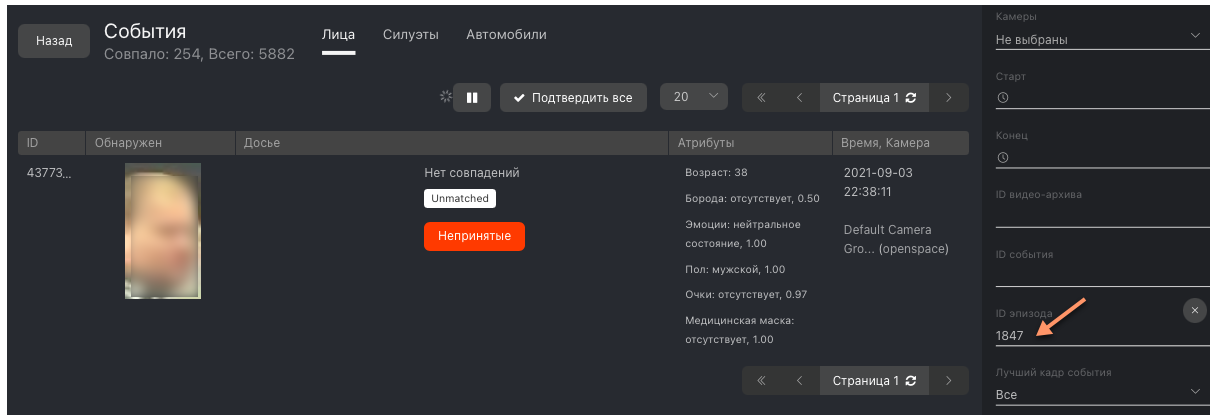
Специальные фильтры для эпизодов с людьми

- *Возраст*: отобразить эпизоды с людьми определенного возраста.
- *Пол*: отобразить эпизоды с людьми заданного пола.
- *Эмоции*: отобразить эпизоды с заданными эмоциями.
- *Очки*: фильтровать эпизоды по наличию очков на лице.
- *Борода*: фильтровать эпизоды по наличию бороды.
- *Liveness*: фильтровать эпизоды по liveness лица.
- *Медицинская маска*: фильтровать эпизоды по наличию медицинской маски.
- *Головной убор*: отобразить только эпизоды с людьми в головном уборе заданного типа: шапка/шляпа/кепка, капюшон/платок, без головного убора.
- *Верх одежды*: отобразить только те эпизоды, в которых человек носит одежду верха заданной обобщенной категории: с длинными рукавами, с короткими рукавами, без рукавов.
- *Тип верха одежды*: отобразить только те эпизоды, в которых человек носит одежду верха заданного типа: куртка, пальто, жилет, толстовка, футболка, рубашка, платье.
- *Цвет верха одежды*: отобразить только эпизоды с людьми, одетыми в одежду заданного цвета (верх).
- *Низ одежды*: отобразить только те эпизоды, в которых человек носит одежду низа заданного типа: брюки, юбка, шорты, неопределенный.
- *Цвет низа одежды*: отобразить только эпизоды с людьми, одетыми в одежду заданного цвета (низ).

Специальные фильтры для эпизодов с автомобилями

- *Тип кузова*: отобразить эпизоды с автомобилями заданного типа кузова.
- *Цвет автомобиля*: отобразить эпизоды с автомобилями заданного цвета.
- *Страна*: отобразить эпизоды с автомобилями, зарегистрированными в заданной стране.
- *Регистрационный номер*: найти эпизод с автомобилем с заданным номером.
- *Регион*: отобразить эпизоды с автомобилями, зарегистрированными в заданном регионе.
- *Марка*: фильтровать эпизоды по марке автомобиля.
- *Модель*: фильтровать эпизоды по модели автомобиля.
- *Спецтранспорт*: отобразить только эпизоды с транспортными средствами, принадлежащими заданному типу: полиция, скорая помощь, спасательные службы, такси, прочие.

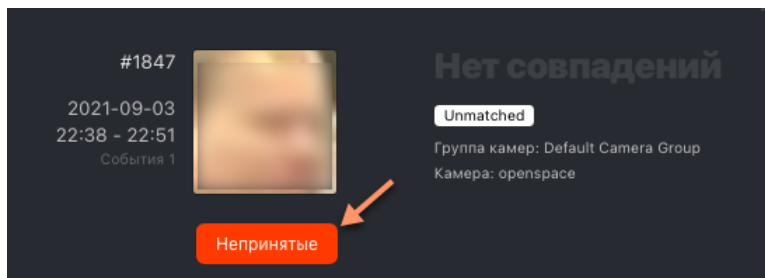
Для просмотра событий эпизода щелкните по нужному эпизоду в списке. Вы будете переправлены на вкладку *События* с соответствующим ID эпизода в фильтре *Эпизод*:



Работа с вкладкой *События* описана в разделе *Работа с событиями*.

Принятие события и эпизода

Для того чтобы подтвердить эпизод целиком, нажмите *Непринятые* в списке эпизодов. В результате все события в эпизоде будут автоматически подтверждены, включая события, которые еще не добавлены (в случае эпизода LIVE).



Эпизод также автоматически подтверждается, если вы приняли все события по отдельности.

Фильтрация событий по ID эпизода

Для того чтобы отобразить события по ID эпизода, используйте фильтр *id* на вкладке *Эпизоды* или фильтр *ID эпизода* на вкладке *События*.

Параметры эпизода

Для настройки эпизодов вам понадобится файл конфигурации `/etc/findface-security/config.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 /etc/findface-security/config.py

...

FFSECURITY = {
    ...
    # episodes specific matching threshold that is used to join faces in an episode
    'FACE_EPISODES_THRESHOLD': 0.688, # FAR = 1.00E-06 # model: [kiwi_320]
    'BODY_EPISODES_THRESHOLD': 0.61, # model: [andariel]
    'CAR_EPISODES_THRESHOLD': 0.61, # model: [alonso]
    ...
    'EPISODE_KEEP_ONLY_BEST_EVENT': True,
    ...
    # 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,
    ...
}

...
```

3.8 Идентификация объектов в офлайн видео

Помимо идентификации объектов на видео в реальном времени, FindFace Multi также позволяет выполнять идентификацию объектов в офлайн видео. Данная функциональность имеет широкий спектр возможных применений, среди которых наиболее распространенным случаем является обнаружение и распознавание объектов в архивных видео.

В этой главе:

- *Форматы видеофайлов*
- *Обработка видеофайла*
- *Настройка обработки видеофайлов*

3.8.1 Форматы видеофайлов

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

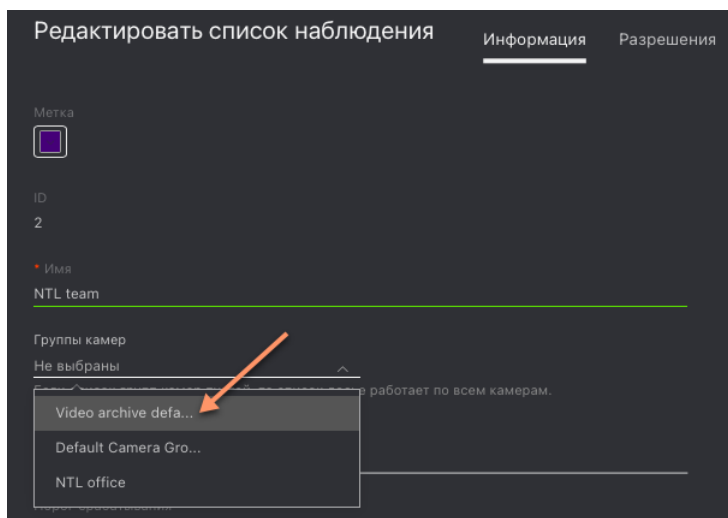
Инсталляции с ускорением CPU и GPU поддерживают все кодеки FFmpeg. Кроме того, поддерживаются следующие кодеки:

- *Ускорение на базе CPU*: flv (как кодек, так и как контейнер), h263, h264, h265, mjpeg, vp8, vp9, mpeg1video, mpeg2video, msmpeg4v2, msmpeg4v3
- *Ускорение на базе GPU*: mjpeg, h264, h265, vp9 и другие, в зависимости от списка кодеков, поддерживаемых используемой видеокартой. Кроме того, для инсталляций с GPU-ускорением можно включить ускорение на основе CPU, тем самым добавив поддержку flv, которая недоступна по умолчанию.

3.8.2 Обработка видеофайла

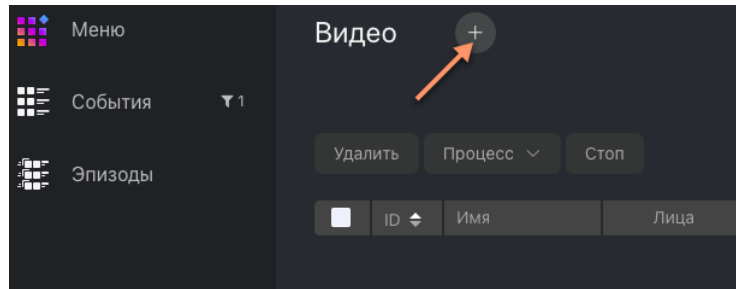
Важно: Перед обработкой видеофайла необходимо выполнить следующие подготовительные действия:

1. Назначьте группу камер, к которой система будет приписывать события с объектами из данного видео. Группа камер **Video archive default** идеально подходит для этой задачи. Вы также можете создать новую группу камер с базовыми настройками специально для данного видео.
2. Укажите данную группу камер в настройках тех *списков наблюдения*, объекты в которых нужно найти на видео.

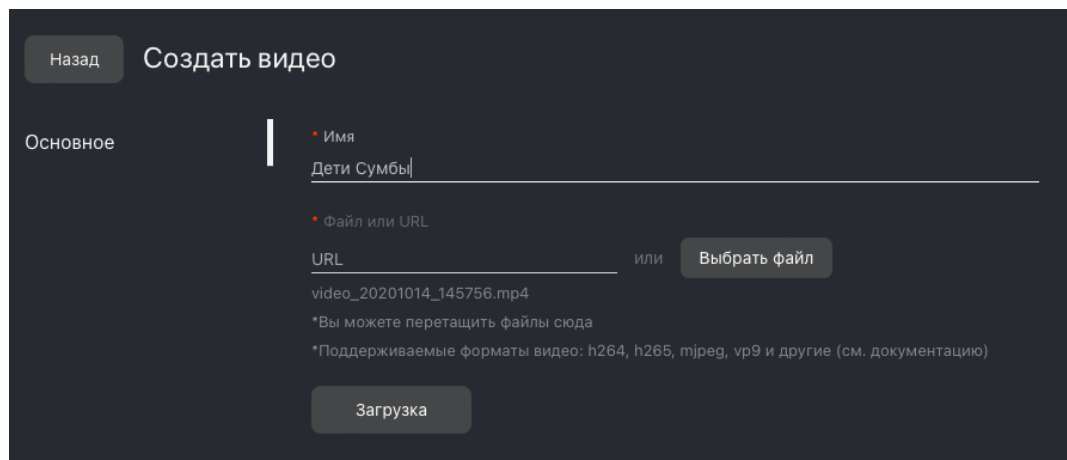


Для идентификации объекта в офлайн видео выполните следующие действия:

1. Создайте видео в FindFace Multi, загрузив его из файла или онлайн-хранилища/облака. Для этого, перейдите на вкладку *Видео*.
2. Нажмите **+**.



3. Укажите название видео.



4. Укажите URL видеоизображения в онлайн-хранилище или выберите видеофайл.
5. Нажмите *Загрузка*.
6. По завершении загрузки видео перейдите на вкладку *Параметры*. Задайте параметры обработки видео:

Назад Редактировать видео

Основное

Параметры

Дополнительное

Аналитика

Область слежения

Области интереса

Обработка

Группа камер
Video archive defa... ▾

Камера
Не выбраны ▾

Отметка времени

Передавать на сервер временные метки из видеопотока (use_stream_timestamp)
☐ false

Если true - отправлять на сервер временные метки, полученные из потока. Если false - отправлять текущие дату и время. По умолчанию передается текущие дата и время.

Указать временную метку для загружаемого видео
🕒 1970-01-01 05:00:00 Сейчас

Чтобы самостоятельно указать временную метку для загружаемого видео, включите выше передачу на сервер временных меток из видеопотока.

Сохранить

- Укажите назначенную ранее группу 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.

7. На вкладке *Дополнительное* выполните тонкую настройку обработки видеофайла:

Назад
Редактировать видео

Основное
Параметры
Дополнительное
Аналитика
Область слежения
Области интереса
Обработка

Объекты распознавания

Лица
☒ true

Силуэты
☒ true

Автомобили
☐ true

Вращение потока

F

Отправка объектов

Время ожидания ответа на запрос (router_timeout_ms)

15000

 15000

Время ожидания в миллисекундах ответа на отправленный запрос с лицом.

Другое

Формат FFMPEG (ffmpeg_format)

Передать FFMPEG формат(mpeg, flv, и т. д.), если он не может быть автоматически определен.

Опции ffmpeg (ffmpeg_params)

Ограничение скорости проигрывания (play_speed)

-1

 -1

Если меньше нуля, то скорость не ограничивается, в остальных случаях поток читается со скоростью play_speed. Не применимо для потоков с камер видеонаблюдения.

Минимальная интенсивность движения (imotion_threshold)

0

 0

Минимальная интенсивность движения, которая будет регистрироваться детектором движения.

Проверять SSL-сертификат (router_verify_ssl)
☒ true

Если true, проверять SSL-сертификат сервера при отправке на него объектов через https. Если false, может быть принят самоподписанный сертификат.

Сохранить

- Поставьте флажки для тех детекторов, которые нужно использовать при обработке видео: лица, силуэты, автомобили.
- При необходимости измените ориентацию видео.
- *Время ожидания ответа на запрос*: Таймаут в миллисекундах отправки на сервер обнаруженных объектов.
- *Формат FFMPEG*: Передать формат FFMPEG (mpeg, flv, и т. д.), если он не может быть автоматически определен.
- *Опции FFMPEG*: Опции FFMPEG для видеопотока. Задаются массивом строк ключ=значение, например, "rtsp_transport=tcp ss=00:20:00".
- *Ограничение скорости проигрывания*: Если меньше нуля, то скорость не ограничивается, в остальных случаях поток читается с play_speed скоростью.

- *Минимальная интенсивность движения*: Минимальная интенсивность движения, которая будет регистрироваться детектором движения.
- *Проверить SSL-сертификат*: Поставьте флажок, чтобы проверять SSL-сертификат сервера при отправке на него объектов через https. Снимите флажок, если вы используете самоподписанный сертификат.

8. На вкладке *Аналитика* укажите настройки детектора объектов каждого типа.

Назад
Редактировать видео

Основное
Лица
Силуэты

Параметры
Дополнительное
Аналитика
Область слежения
Области интереса
Обработка

Минимальное качество изображения объекта (filter_min_quality)
0.45
0.45

Минимальное качество изображения лица для детекции.
Допустимый диапазон от 0 до 1.
Базовое рекомендуемое значение 0.45 - соответствует удовлетворительному качеству изображения лица.
Значения вблизи 1 соответствуют наиболее качественным выровненным изображениям лиц, полностью находящихся в кадре.
Значения вблизи 0 соответствуют замыленным изображениям, а также изображениям, содержащим перевернутые лица или лица, повернутые под большими углами. При таких значениях распознавание может быть неэффективным.

Минимальный размер объекта (filter_min_size)
1
1

Минимальный размер объекта в пикселях для отправки на сервер.

Максимальный размер объекта (filter_max_size)
8192
8192

Максимальный размер объекта в пикселях для отправки на сервер.

Качество сжатия (jpeg_quality)
95
95

Качество сжатия полного кадра для отправки.

Буферный режим (overall_only)
☐ false

Буферный режим. Отправлять для объекта один кадр наилучшего качества.

Временной интервал (realtime_post_interval)
1
1

Временной интервал в секундах (целое или рациональное число), в течение которого в режиме реального времени выбирается лучший кадр с объектом.

Отправлять первый объект сразу (realtime_post_first_immediately)
☐ false

Если true, 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. Если false, отправлять первый объект после окончания первого realtime_post_interval.

Отправлять лучший кадр (realtime_post_every_interval)
☐ false

Если true, отправлять лучший кадр в каждом временном интервале realtime_post_interval в режиме реального времени. Если false, отправлять лучший кадр, только если его качество улучшилось по сравнению с предыдущим отправленным кадром.

Обрезать полный кадр по ROT (fullframe_crop_rot)
☐ false

Если true, то обрезать полный кадр по размеру ROT-области перед отправкой на распознавание. Размер полного кадра будет равен размеру ROT-области.

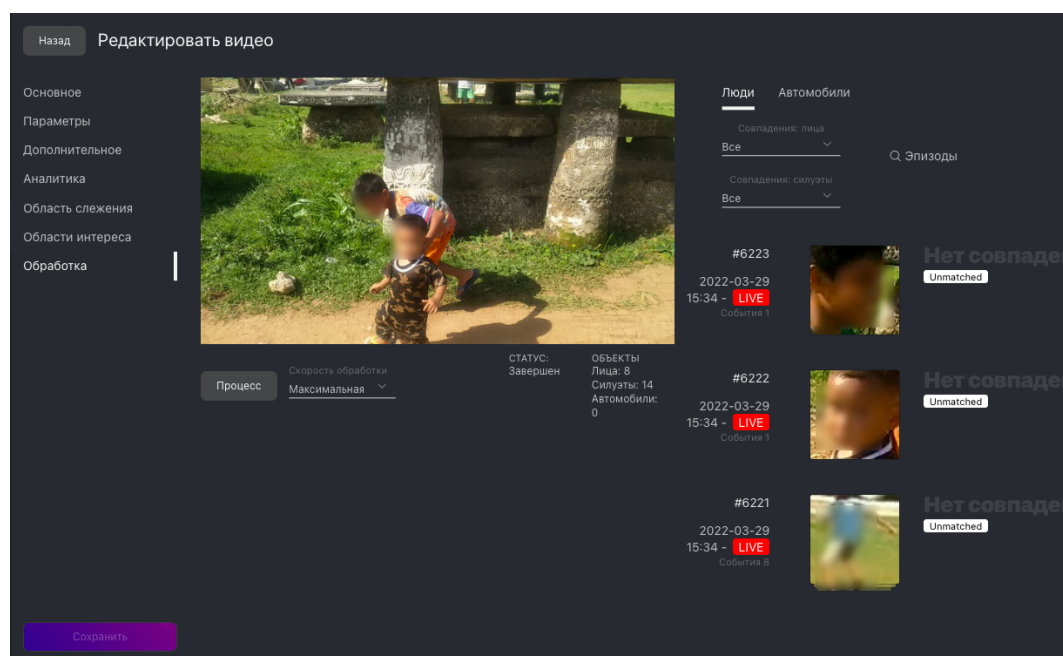
Дополнительные параметры

Сохранить

- *Минимальное качество изображения объекта*: Минимальное качество изображения объекта для отправки на сервер. Не меняйте значение по умолчанию без предварительной консуль-

тации с нашими техническими специалистами (support@ntechlab.com).

- *Минимальный размер объекта*: Минимальный размер объекта в пикселях для отправки на сервер.
 - *Максимальный размер объекта*: Максимальный размер объекта в пикселях для отправки на сервер.
 - *Качество сжатия*: Качество сжатия полного кадра для отправки.
 - *Буферный режим*. Отправлять для объекта один кадр наилучшего качества.
 - *Временной интервал*: Временной интервал в миллисекундах (целое или десятичное число), в течение которого в режиме реального времени выбирается лучший кадр с объектом.
 - *Отправлять первый объект сразу*: Поставьте флажок, чтобы отправлять первый объект из трека сразу, как только фильтры по качеству/размеру/региону интереса пройдены, не дожидаясь окончания первого **Временного интервала**. Способ отправки последующих изображений объекта зависит от установленного значения **Отправлять лучший кадр**. Снимите флажок, чтобы отправлять первый объект из трека только после окончания первого **Временного интервала**.
 - *Отправлять лучший кадр*: Поставьте флажок, чтобы отправлять лучший кадр в каждом **Временном интервале** в режиме реального времени. Снимите флажок, чтобы отправлять лучший кадр, только если его качество улучшилось по сравнению с предыдущим отправленным кадром.
9. (Опционально) На вкладках *Область слежения* и *Области интереса* укажите регион отслеживания объектов в поле зрения камеры и зоны детекции для каждого типа объектов.
10. Перейдите на вкладку *Обработка*. Нажмите *Процесс*, чтобы запустить идентификацию объектов.



Вы можете просмотреть события идентификации объектов прямо здесь, а также на вкладках *События* и *Эпизоды*, отфильтровав список событий по группе камер/ камере, ассоциированной с видео.

3.8.3 Настройка обработки видеофайлов

По умолчанию видеофайлы обрабатываются в режиме очереди для предотвращения пропуска событий из-за чрезмерного потребления ресурсов. Вы можете изменить количество одновременно обрабатываемых видеофайлов, заданное по умолчанию. Для этого откройте файл конфигурации `/etc/findface-security/config.py` и измените параметр `MAX_VIDEO_ARCHIVE_JOBS`. Пожалуйста, предварительно свяжитесь с нашими экспертами (support@ntechlab.com), чтобы убедиться, что ресурсов достаточно.

```
sudo vi /etc/findface-security/config.py

...

FFSECURITY = {
    ...
    # maximum concurrent video manager jobs for video archives processing
    'MAX_VIDEO_ARCHIVE_JOBS': 3,
    ...
}
...
```

После внесения изменений обязательно перезапустите сервис `findface-security`.

```
sudo systemctl restart findface-security.service
```

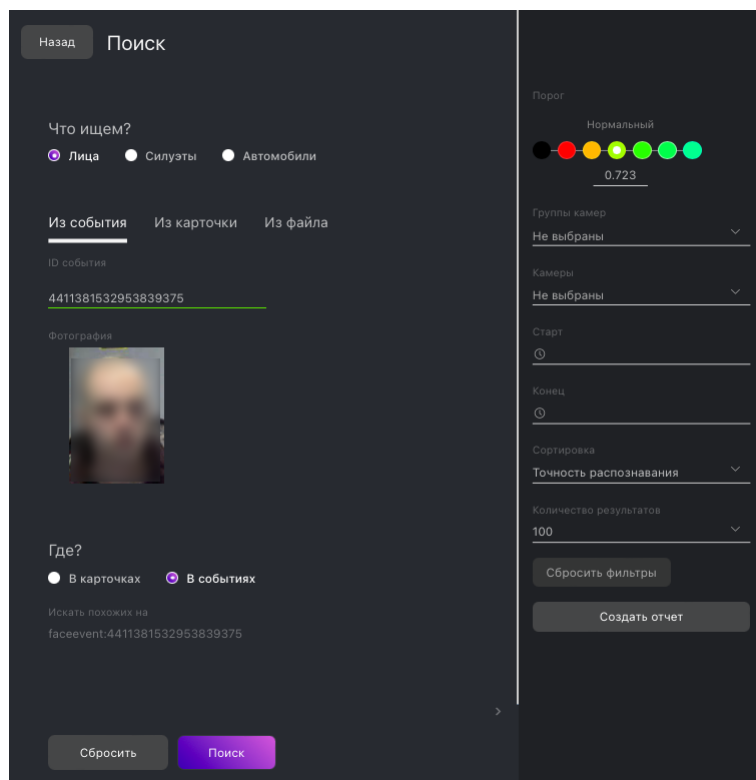
3.9 Поиск объектов в базах данных

FindFace Multi позволяет искать объекты в следующих базах данных:

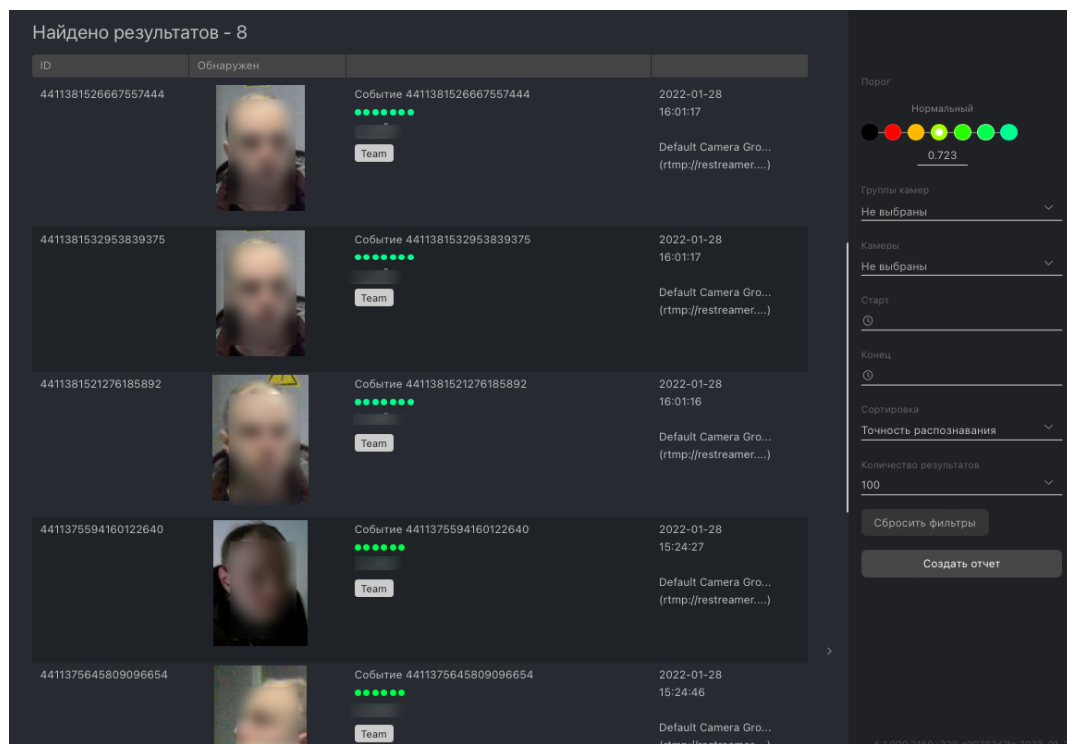
- База данных обнаруженных объектов: вкладка *События*.
- Картотека: вкладка *Карточки*.

Для поиска объекта в базе данных выполните следующие действия:

1. Перейдите на вкладку *Поиск*.



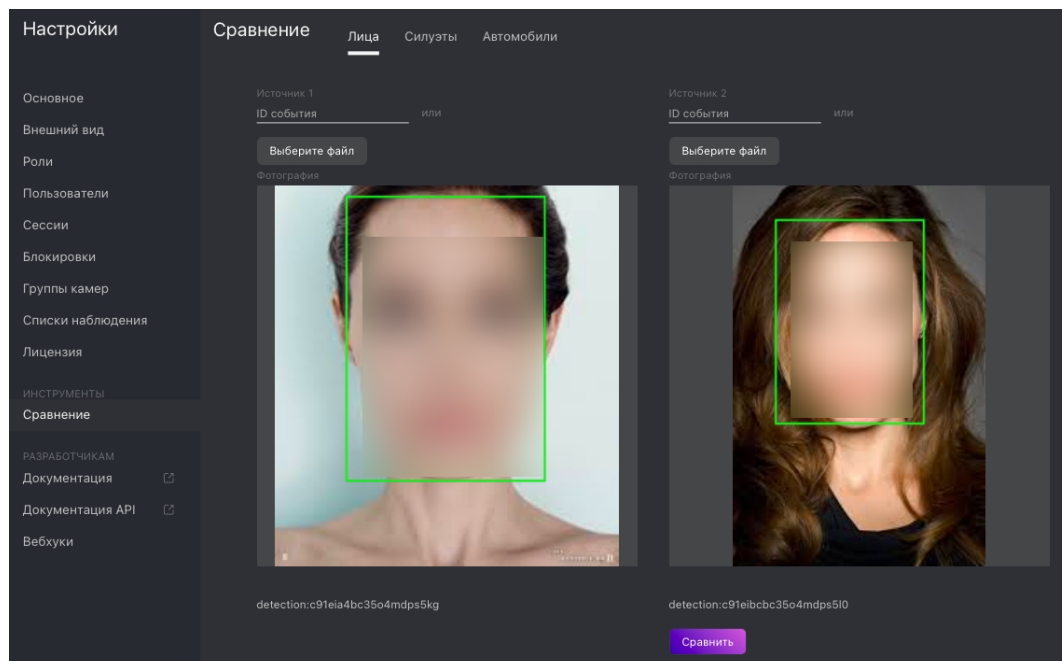
2. Выберите тип искомых объектов: *Лица, Силуэты, Автомобили.*
3. Укажите базу данных для поиска: *В событиях, В карточках.*
4. Задайте искомый объект одним из следующих способов:
 - ID события.
 - ID карточки. Если карточка содержит несколько фотографий, выберите те, которые будут использованы для поиска.
 - Загрузив фотографию. Фотография будет отображена в одноименном поле. Если на фотографии присутствует несколько объектов, выберите нужный.
5. Укажите максимальное количество событий в результатах поиска.
6. При поиске в событиях можно сузить область поиска, указав интересующую вас камеру, группу камер и период времени.
7. Нажмите *Поиск*. Результаты поиска будут отображены ниже. Для каждого найденного объекта будет указана вероятность его совпадения с заданным.



3.10 Сравнение двух объектов

FindFace Multi позволяет сравнить два объекта и убедиться в их совпадении. Выполните следующие действия:

1. Перейдите на вкладку *Настройки*. Нажмите *Сравнение*. Выберите нужную вкладку, в зависимости от типа объектов, которые вы собираетесь сравнить: *Лица*, *Силуэты*, *Автомобили*.



2. Укажите id событий, объекты из которых нужно сравнить, и/или загрузите фотографии с объектами.

Совет: Узнать ID события можно на вкладке *События*.

3. Нажмите *Сравнить*. В результате будет отображена вероятность совпадения объектов.

3.11 Кластеры лиц, силуэтов, автомобилей

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 автомобиля образуют кластер автомобилей.

Совокупные галереи кластеров лиц, силуэтов и автомобилей доступны на вкладке *Кластеры*.

Примечание: Если для кластера лиц/силуэтов найдено совпадение с *карточкой человека*, он автоматически отобразится в данной карточке. По аналогии, автомобильный кластер будет сохранен в соответствующей карточке автомобиля.

Важно: По умолчанию кластеризация объектов отключена. *Включите и настройте ее* через файл конфигурации `/etc/findface-security/config.py`.

В этом разделе:

- *Алгоритмы кластеризации*
- *Включение и настройка кластеризации*
- *Работа с галереями кластеров*
 - *Просмотр кластеров. Фильтры*
 - *Объединение и удаление кластеров*
 - *Просмотр связанных кластерных событий. Фильтры*
- *Статическая галерея кластеров*
- *Ручная кластеризация*

3.11.1 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 `/etc/findface-security/config.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*).

3.11.2 Включение и настройка кластеризации

По умолчанию кластеризация отключена. Чтобы включить и настроить ее, выполните следующие действия:

1. Откройте файл конфигурации `/etc/findface-security/config.py`. Найдите раздел `Clusters configuration`.

```
sudo vi /etc/findface-security/config.py

# -- Clusters configuration --
'ENABLE_NIGHT_CLUSTERIZATION': True,
'ENABLE_REALTIME_CLUSTERIZATION': True,
# rrule (recurrence rule) for scheduling clusters 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','body','car'],
  # skip clusterization if unpinned cluster events count is greater than this
→value
  'CLUSTERIZATION_MAX_CLUSTER_EVENTS': None,
  # cluster event to cluster matching confidence threshold
  'FACE_CLUSTER_CONFIDENCE_THRESHOLD': 0.723, # model: [kiwi_320]
  'BODY_CLUSTER_CONFIDENCE_THRESHOLD': 0.65, # model: [andariel]
  'CAR_CLUSTER_CONFIDENCE_THRESHOLD': 0.65, # model: [alonso]
  # minimum required event quality for cluster creation
  'FACE_CLUSTER_EVENT_MIN_QUALITY': 0.45, # model: [quality.v1]
  'BODY_CLUSTER_EVENT_MIN_QUALITY': 0.6, # model: [pedattr.quality.v0]
  'CAR_CLUSTER_EVENT_MIN_QUALITY': 0.73, # model: [carattr.quality.v0]
  # 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,
  # coefficient of dependence of the clustering threshold on the person event's
→quality
  'FACE_CLUSTER_SOFT_CLUSTERIZATION_COEFFICIENT': 0.1,
  'BODY_CLUSTER_SOFT_CLUSTERIZATION_COEFFICIENT': 0.1,
  'CAR_CLUSTER_SOFT_CLUSTERIZATION_COEFFICIENT': 0.1,
  # match with clusters when no card objects matches found
  'MATCH_CLUSTERS': False,
  # create auto cards for unmatched clusters
  'CREATE_AUTO_CARDS': False,
  # 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 then
      'max_centroid_similarity_threshold': {'enabled': True, 'value': 0.55},
      # minimum average event's quality
      'min_average_events_quality': {'enabled': True, 'value': 0.45},
    },
    'body': {},
    'car': {},
  },
  # 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 вклю-

чена 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 автомобилей, 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': 0.45, # model: [quality.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': 0.723, # model: [kiwi_320]
'BODY_CLUSTER_CONFIDENCE_THRESHOLD': 0.65, # model: [andariel]
'CAR_CLUSTER_CONFIDENCE_THRESHOLD': 0.65, # model: [alonso]
```

9. После обнаружения объекта на видеоизображении система ищет в картотеке совпадающие с ним объекты. Установите `'MATCH_CLUSTERS': True`, чтобы система могла искать совпадения в галерее кластеров, если она не нашла подходящей карточки в картотеке. В этом случае система будет уведомлять вас всякий раз, когда объект совпадает с ранее созданным кластером.

```
# match with clusters when no card objects matches found
'MATCH_CLUSTERS': True,
```

10. Установите `'CREATE_AUTO_CARDS': True`, чтобы автоматически создавать новые карточки для кластеров, у которых отсутствуют совпадения в картотеке. Подробнее см. *Автокарточки и заполненные карточки*.

```
# create auto cards for unmatched clusters
'CREATE_AUTO_CARDS': True,
```

11. 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/автомобилю. 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': {
    # 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': 0.55},
    # 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,
```

12. 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,
```

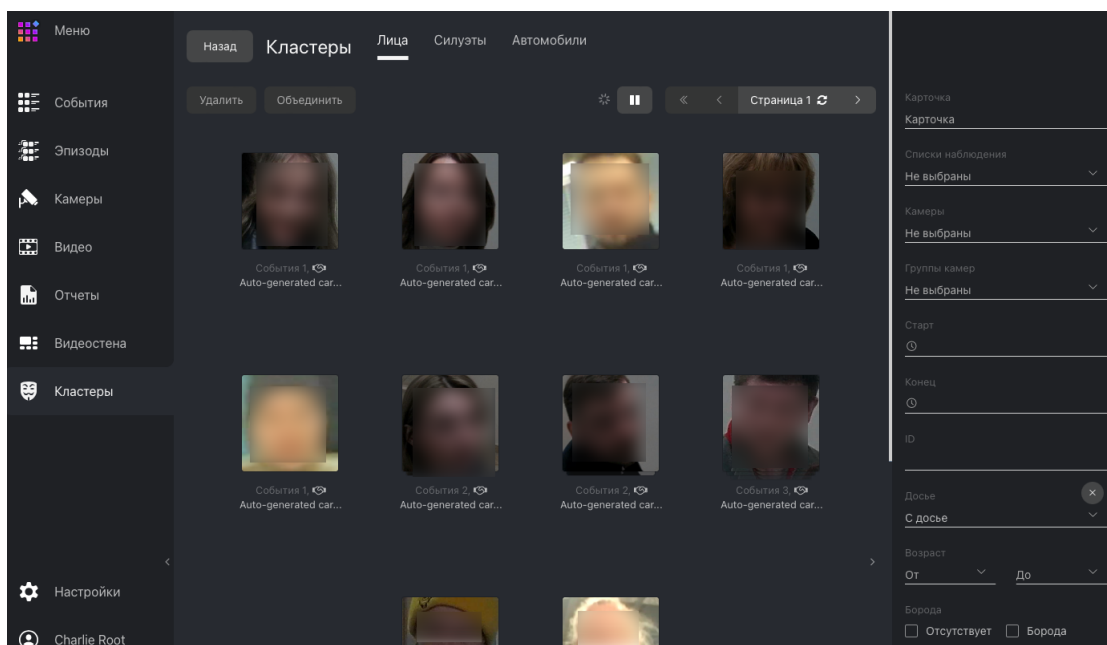
13. Перезапустите сервис findface-security. В веб-интерфейсе FindFace Multi появится вкладка *Кластеры*.

```
sudo systemctl restart findface-security.service
```

3.11.3 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 активированного функционала распознавания.

- *Карточка*: отобразить только кластеры по определенной карточке.
- *Камеры*: отобразить только кластеры по определенной камере.
- *Группы камер*: отобразить только кластеры по определенной группе камер.
- *Списки наблюдения*: отобразить только кластеры по определенному списку наблюдения.
- *Старт, Конец*: отобразить только кластеры, сформированные в определенный период времени.
- *ID*: отобразить кластер с определенным ID.
- *Досье*: отобразить кластеры в зависимости от того, имеют ли они связанную карточку, автоматически сгенерированную или заполненную (подробное объяснение см. в разделе [Автокарточки и заполненные карточки](#)).

Специальные фильтры для кластеров лиц

- *Возраст*: отобразить кластеры с людьми определенного возраста.
- *Пол*: отобразить кластеры с людьми заданного пола.
- *Эмоции*: отобразить кластеры с заданными эмоциями.
- *Очки*: фильтровать кластеры по наличию очков на лице.
- *Борода*: фильтровать кластеры по наличию бороды.
- *Liveness*: фильтровать кластеры по liveness лица.
- *Медицинская маска*: фильтровать кластеры по наличию медицинской маски.

Специальные фильтры для кластеров силуэтов

- *Головной убор*: отобразить только кластеры с людьми в головном уборе заданного типа: шапка/шляпа/кепка, капюшон/платок, без головного убора.
- *Верх одежды*: отобразить только те кластеры, в которых человек носит одежду верха заданной обобщенной категории: с длинными рукавами, с короткими рукавами, без рукавов.
- *Тип верха одежды*: отобразить только те кластеры, в которых человек носит одежду верха заданного типа: куртка, пальто, жилет, толстовка, футболка, рубашка, платье.
- *Цвет верха одежды*: отобразить только кластеры с людьми, одетыми в одежду заданного цвета (верх).
- *Низ одежды*: отобразить только те кластеры, в которых человек носит одежду низа заданного типа: брюки, юбка, шорты, неопределенный.
- *Цвет низа одежды*: отобразить только кластеры с людьми, одетыми в одежду заданного цвета (низ).

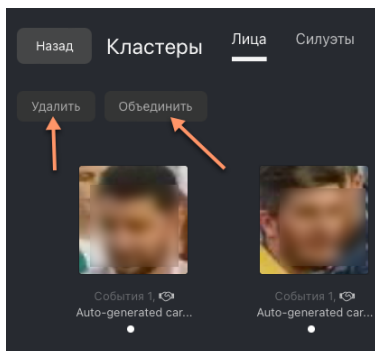
Специальные фильтры для кластеров автомобилей

- *Тип кузова*: отобразить кластеры с автомобилями заданного типа кузова.
- *Цвет автомобиля*: отобразить кластеры с автомобилями заданного цвета.
- *Марка*: фильтровать кластеры по марке автомобиля.
- *Модель*: фильтровать кластеры по модели автомобиля.
- *Спецтранспорт*: отобразить только кластеры с транспортными средствами, принадлежащими заданному типу: полиция, скорая помощь, спасательные службы, такси, прочие.

Объединение и удаление кластеров

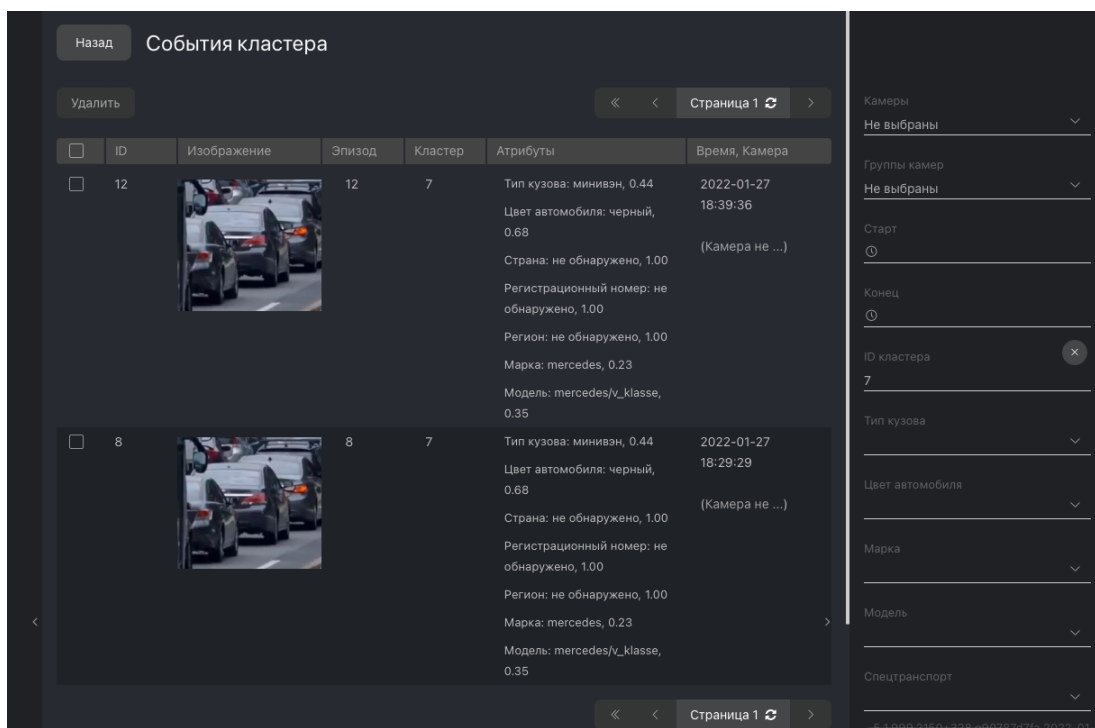
Для того чтобы вручную объединить несколько кластеров, выберите их один за другим и нажмите *Объединить*.

Для того чтобы удалить кластер, выберите его и нажмите *Удалить*.



Просмотр связанных кластерных событий. Фильтры

Щелкните по нужному кластеру, чтобы просмотреть связанные с ним кластерные события. Вы будете перенаправлены на страницу *События кластера* с соответствующим значением в фильтре *ID кластера*.



Набор фильтров для работы с кластерными событиями аналогичен *набору фильтров для кластеров*.

3.11.4 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 `/etc/findface-security/config.py`.

```
sudo vi /etc/findface-security/config.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 сервис findface-security.

```
sudo systemctl restart findface-security.service
```

3.11.5 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 findface-security run_clusterization --help
usage: findface-security 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.
```

(continues on next page)

(продолжение с предыдущей страницы)

<code>--force-color</code>	Force colorization of the command output.
<code>--skip-checks</code>	Skip system checks.

Данная утилита позволяет по отдельности запускать кластеризацию лиц, силуэтов и автомобилей, 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 findface-security run_clusterization --object-types face --force
```

См.также:

- [Файл конфигурации findface-security](#)
- [Настройка мониторинга объектов и картотеки](#)
- [Анализ взаимодействий](#)
- [Аналитика «Знай своего клиента» \(KYC\)](#)

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.

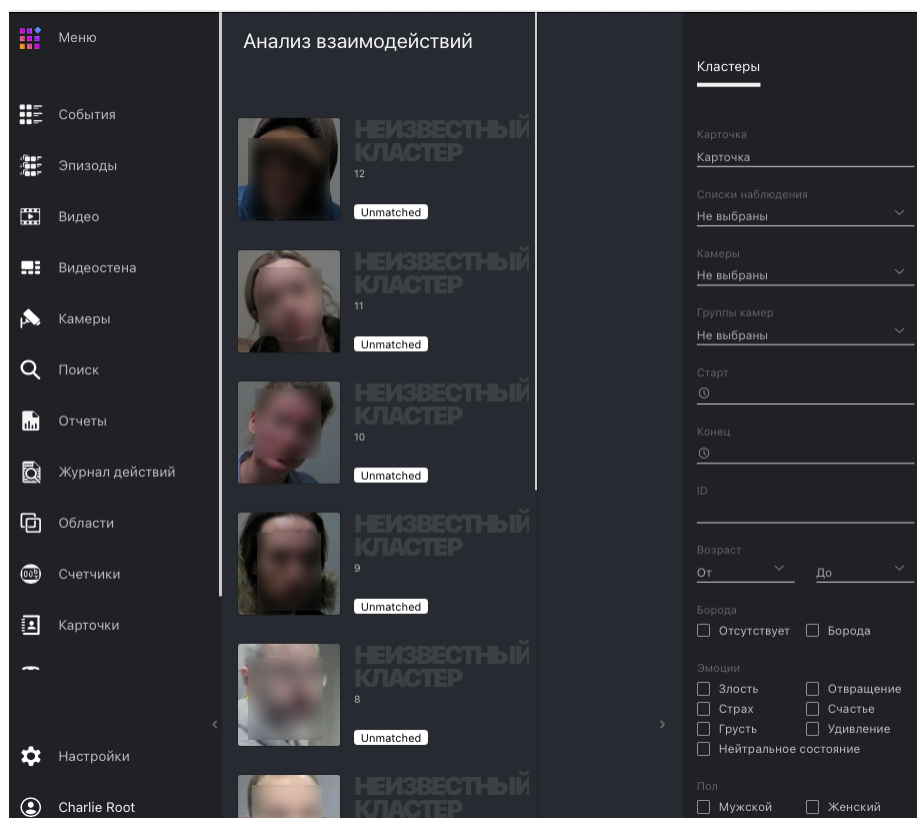
3.12.1 Анализ взаимодействий

FindFace Multi может выполнять отслеживание и анализ взаимодействий. Для использования данного функционала вам потребуется включить *кластеризацию лиц*.

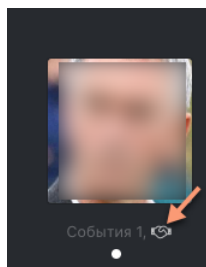
При отслеживании взаимодействий система автоматически определяет круг людей, с которыми человек ранее контактировал. Для каждого человека из первого круга система строит еще один круг ассоциированных людей и так далее. В целом, анализ взаимодействий состоит из трех кругов.

Примечание: Вы также можете отслеживать взаимодействия между силуэтами и автомобилями. Данная функция доступна через [HTTP API](#). Обратите внимание, что кластеризация силуэтов и автомобилей должна быть предварительно включена.

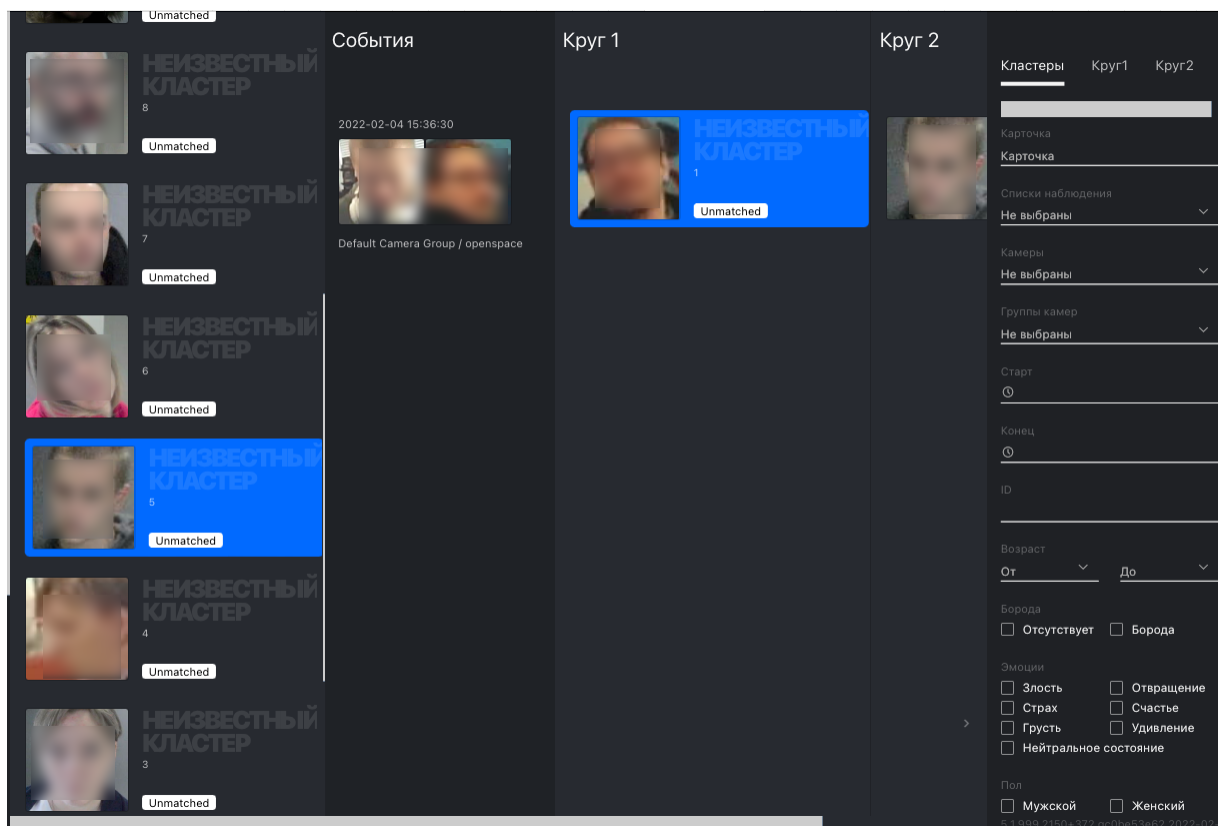
Анализ взаимодействий доступен на вкладке *Анализ взаимодействий*.



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.

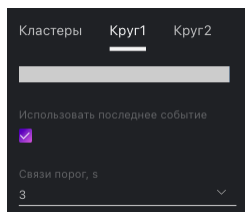


Фильтры можно применять к каждому кругу.

Совет: Например, вы можете найти всех пожилых людей или людей без маски, кто прямо или косвенно взаимодействовал с потенциально инфицированным человеком.

Совет: При поиске по кругу связей применяются следующие настройки:

- *Использовать последнее событие:* 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.
- *Связи порог:* максимальное время в секундах между появлением в кадре людей, чтобы считать их связанными.

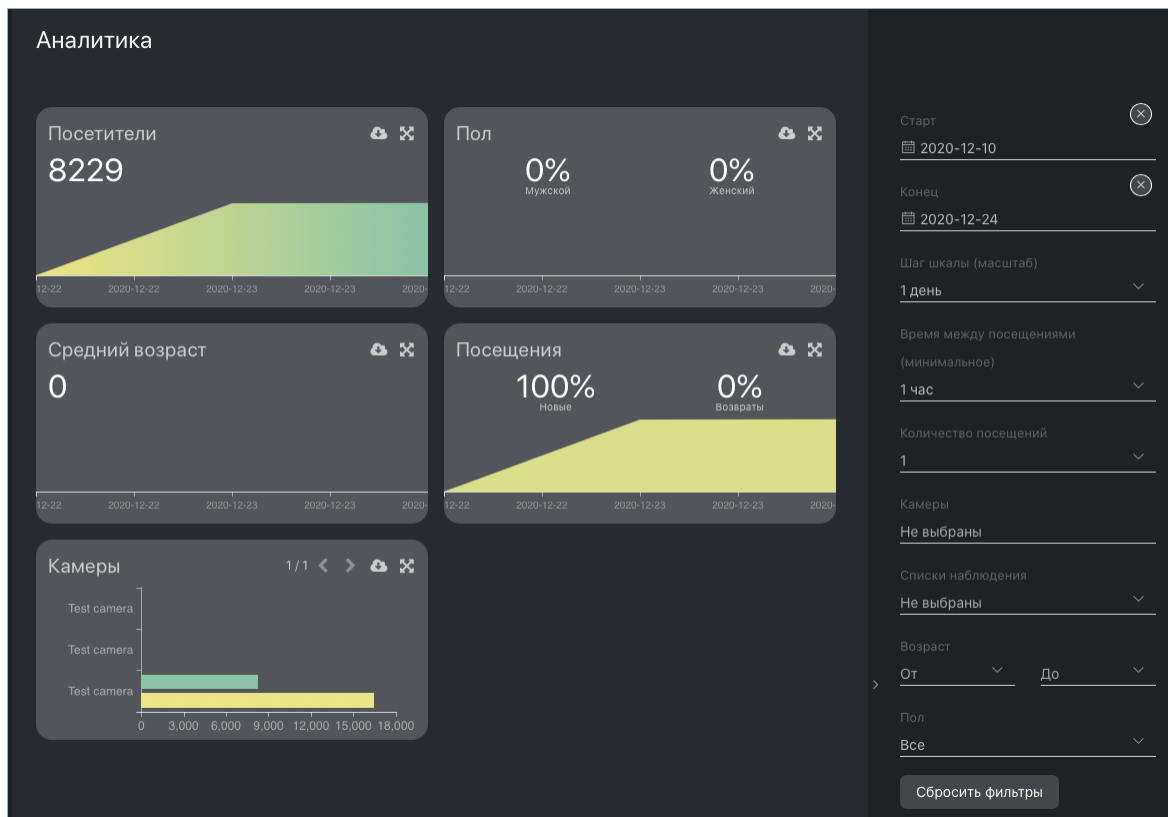


3.12.2 Аналитика «Знай своего клиента» (KYC)

Аналитика «Знай своего клиента» (Know Your Customer, KYC) 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). Это отличный стартовый инструмент, позволяющий внедрить в бизнес принципы KYC.

Диаграммы с аналитическими данными доступны на вкладке *Аналитика*.

Важно: Построение аналитики выполняется только при включенной *кластеризации лиц*.



Для работы с данными аналитики используйте следующие фильтры:

- Период времени
- Шаг шкалы
- Время между посещениями
- Количество посещений
- Камеры
- Списки наблюдения
- Возраст
- Пол

См. также:

- *Кластеры лиц, силуэтов, автомобилей*

3.13 Отчеты

В этой главе:

- *Типы отчетов*
- *Построение стандартного отчета*
- *Построение отчета о рабочем времени*
- *Работа с отчетами*
- *Настройка сохранения изображений в отчетах*

3.13.1 Типы отчетов

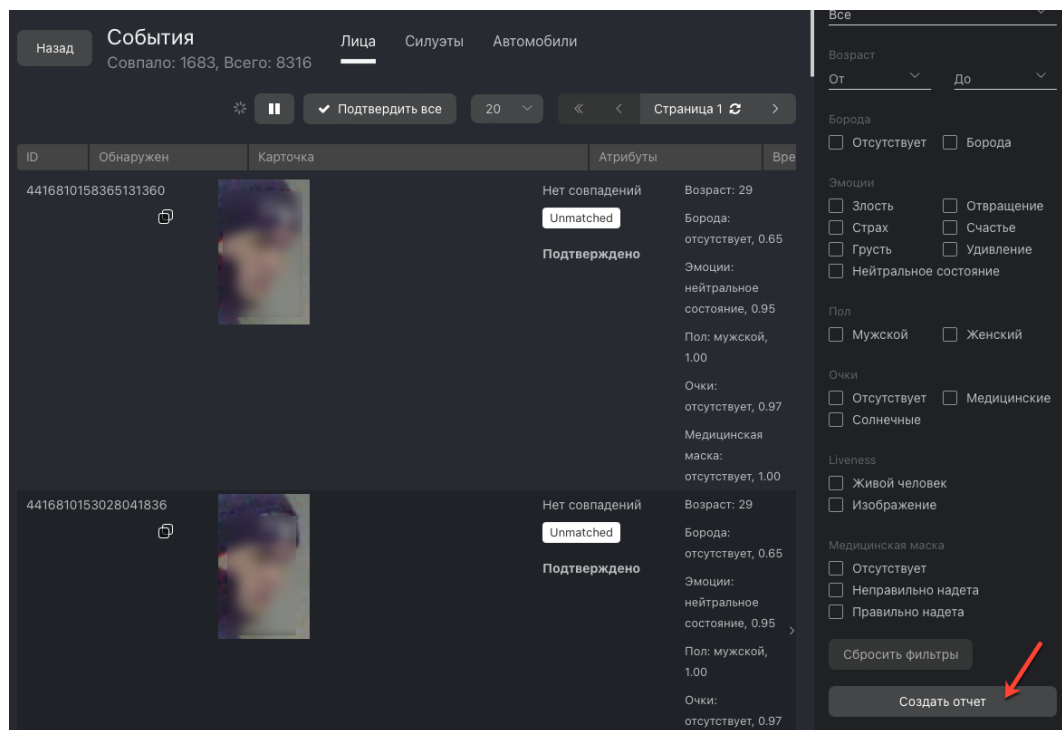
В FindFace Multi доступны следующие отчеты:

- Стандартные отчеты по следующим системным объектам:
 - *события*
 - *эпизоды*
 - *поиск*
 - *кластеры*
 - *камеры*
 - *карточки*
 - *аналитические данные*
 - *журнал действий пользователей*
 - *области*, активации областей, записи активаций областей
- Отчеты о рабочем 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 как на вкладке *События*, так и *Области*.

3.13.2 Построение стандартного отчета

Для того чтобы построить стандартный отчет по системному объекту, выполните следующие действия:

1. Перейдите на вкладку, соответствующую нужному системному объекту: *События*, *Эпизоды*, *Поиск*, *Кластеры*, *Камеры*, *Карточки*, *Аналитика*, *Журнал действий*, *Области*.
2. Установите фильтры для отчета.
3. Нажмите *Создать отчет*.

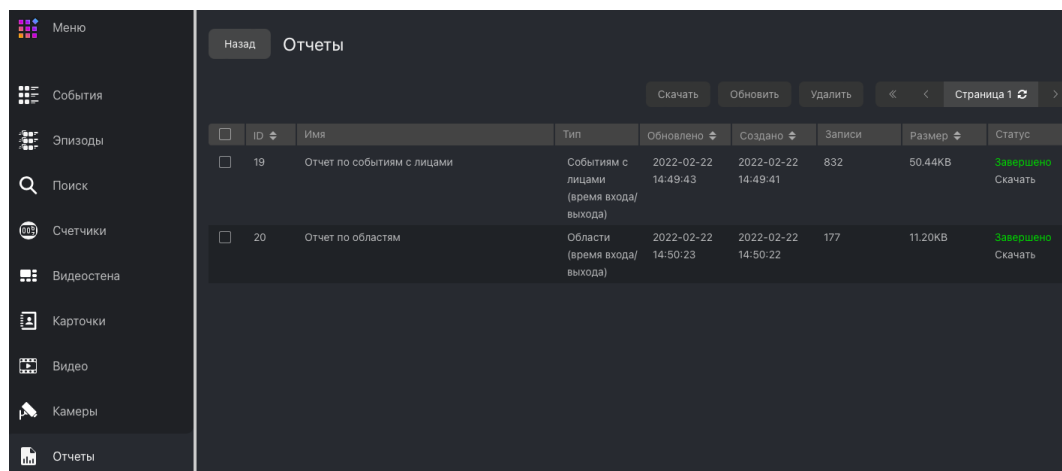


4. Укажите имя отчета.
5. Если вы находитесь на вкладке *События* или *Области*, выберите тип отчета *Стандартный отчет* (поскольку на данных вкладках доступно два типа отчетов).
6. Если применимо, выберите, следует ли сохранять изображения отчета в виде ссылок, миниатюр или полных кадров.

The screenshot shows the 'Создать отчет' (Create report) dialog box. It has a close button (X) in the top right corner. The form includes the following fields and options:

- Имя** (Name): A text input field.
- Отчет по событиям с лицами** (Report on events with faces): A text input field.
- Тип отчета** (Report type): A dropdown menu currently showing 'Стандартный отчет' (Standard report).
- Сохранить изображения в отчете как** (Save images in report as): A dropdown menu.
- Ссылка** (Link): A text input field.
- Создать** (Create): A purple button at the bottom right.

7. Нажмите *Создать*. Отчет будет доступен для загрузки на вкладке *Отчеты*.



3.13.3 Построение отчета о рабочем времени

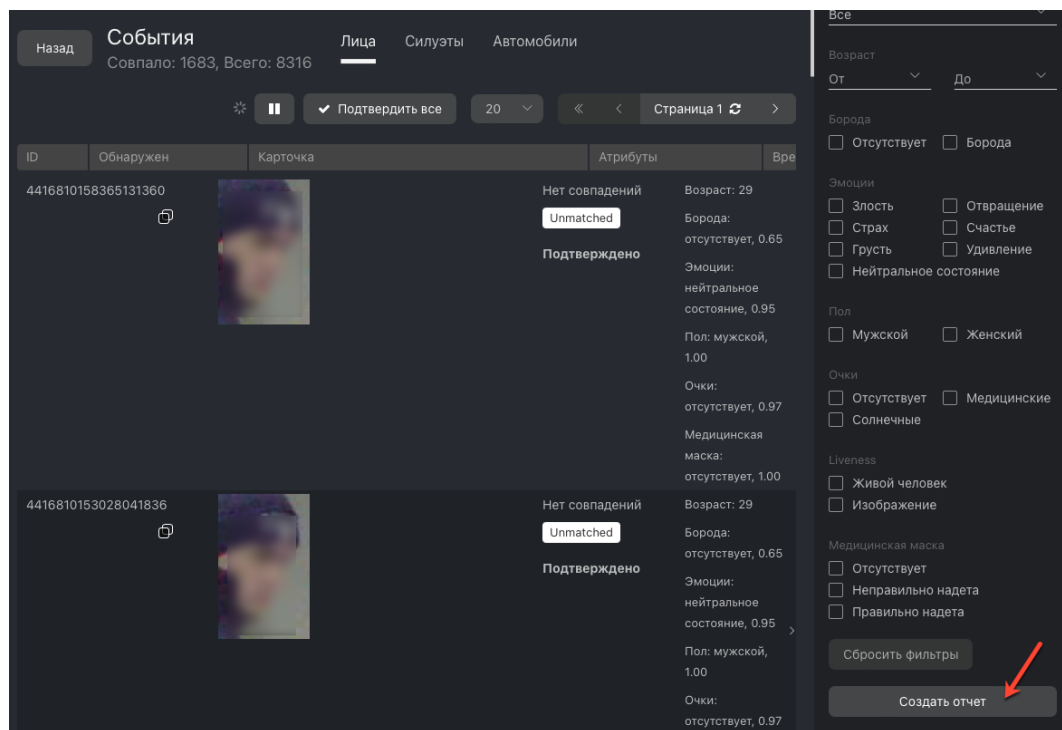
Отчет о рабочем времени можно построить на следующих вкладках:

- *События*: персонифицированный общий отчет о рабочем времени. Требует указания камер входа и выхода.
- *Области*: неперсонифицированный отчет о рабочем времени, касающийся использования определенной области. Фиксирует моменты входа и выхода из области с заданным ID.

События

Для того чтобы построить отчет о рабочем времени на вкладке *События*, выполните следующие действия:

1. Установите фильтры для отчета.
2. Нажмите *Создать отчет*.



3. Укажите имя отчета.
4. Выберите тип отчета: *Учет рабочего времени*.
5. Выберите камеру входа.
6. Выберите камеру выхода.
7. Если применимо, выберите, следует ли сохранять изображения отчета в виде ссылок, миниатюр или полных кадров.

Создать отчет

Имя

Отчет по событиям с лицами

* Тип отчета

Учёт рабочего времени

* Камера вход

openspace

* Камера выход

exit

* Сохранить изображения в отчете как

Ссылка

Создать

8. Нажмите *Создать*. Отчет будет доступен для загрузки на вкладке *Отчеты*.

Области:

Для того чтобы построить отчет о рабочем времени на вкладке *Области*, выполните следующие действия:

1. Установите фильтры для отчета.
2. Нажмите *Создать отчет*.

Назад Области + Области События

Удалить

« < Страница 1 ↻ > »

<input type="checkbox"/>	ID	Изображение	Имя	Создано	Активная
<input type="checkbox"/>	1		1 / Default Camera Group Менее чем 1 человек, длительность 30 секунд	2022-02-17 18:30:27	<input checked="" type="checkbox"/>

« < Страница 1 ↻ > »

Группы камер
Не выбраны

Камеры
Не выбраны

Сбросить фильтры

Создать отчет

3. Укажите имя отчета.
4. Выберите тип отчета: *Учет рабочего времени*.

5. Укажите имя отчета.

Создать отчет

Имя

Отчет по областям

ID области

Тип отчета

Учёт рабочего времени

Создать

6. Нажмите *Создать*. Отчет будет доступен для загрузки на вкладке *Отчеты*.

3.13.4 Работа с отчетами

Вы можете получить доступ к отчетам, ранее созданным в системе, на вкладке *Отчеты*. Доступны следующие операции:

- Скачивание выбранных отчетов в формате XLSX.
- Обновление выбранных отчетов.
- Удаление выбранных отчетов.

Назад Отчеты

Скачать Обновить Удалить < < Страница 1 > >

<input type="checkbox"/>	ID	Имя	Тип	Обновлено	Создано	Записи	Размер	Статус
<input type="checkbox"/>	19	Отчет по событиям с лицами	Событиям с лицами (время входа/выхода)	2022-02-22 14:49:43	2022-02-22 14:49:41	832	50.44KB	Завершено Скачать
<input type="checkbox"/>	20	Отчет по областям	Области (время входа/выхода)	2022-02-22 14:50:23	2022-02-22 14:50:22	177	11.20KB	Завершено Скачать

3.13.5 Настройка сохранения изображений в отчетах

При создании некоторых 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 `/etc/findface-security/config.py` и измените заданные по умолчанию качество JPEG и максимальную высоту миниатюр и полных кадров в зависимости от свободного дискового пространства.

```
sudo vi /etc/findface-security/config.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,
```

3.14 Видеостена

В FindFace Multi встроен базовый функционал видеонаблюдения. Используйте видеостену для отображения видео с камер и видеофайлов.

Важно: Расширенный функционал видеонаблюдения предоставляется функцией *Видеомониторинг*.

В этой главе:

- *Отображение видео*
- *Настройка видеостены*

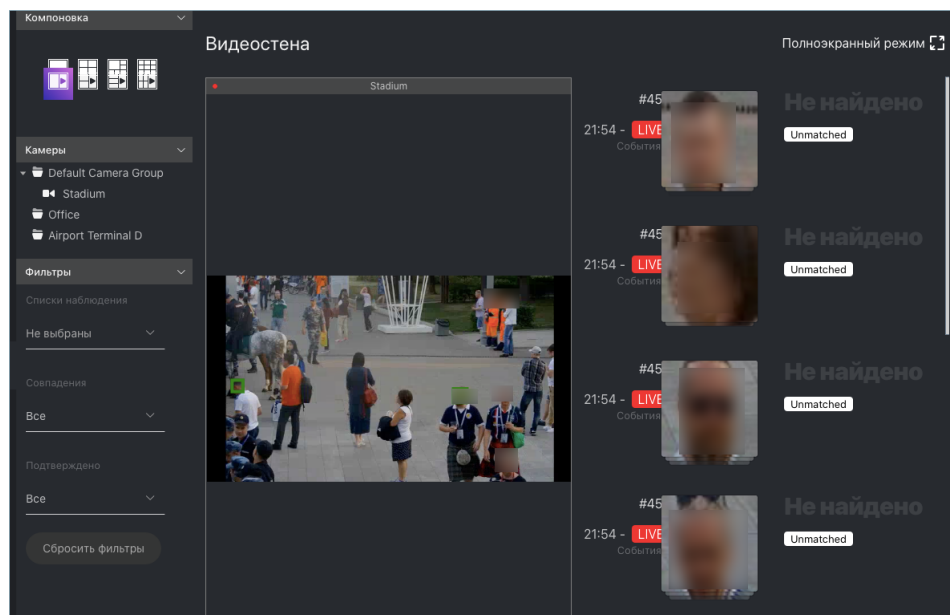
3.14.1 Отображение видео

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 на видеостене *по аналогии* с вкладкой *Эпизоды*, включая следующие основные фильтры:

- *Списки наблюдения*
- *Совпадения*
- *Подтверждено*

3.14.2 Настройка видеостены

Вы можете настроить функционал видеостены, определяющий способ выделения объектов на видео:

- включить/выключить отображение рамок и данных по атрибутам для лиц, автомобилей и силуэтов

Примечание: Данные по атрибутам отображаются, если вы включили нейронные сети для распознавания атрибутов. См. *Распознавание атрибутов лица*, *Распознавание автомобилей и их атрибутов*, *Распознавание силуэтов и их атрибутов*.

- включить размытие объектов без совпадений в целях соблюдения законов о защите персональных данных (GDPR и аналогичных)

Для настройки видеостены выполните следующие действия:

1. Откройте файл конфигурации `/etc/findface-security/config.py`.

```
sudo vi /etc/findface-security/config.py
```

2. Найдите секцию `FFSECURITY_UI_CONFIG -> available_video_wall_features`.
3. Установите `"faces"`, `"cars"`, `"bodies"` `True` или `False`, чтобы включить/выключить отображение рамок и данных по атрибутам для лиц, автомобилей и силуэтов соответственно.

```
FFSECURITY_UI_CONFIG = {  
    ...  
    "available_video_wall_features": {  
        "faces": True,  
        "cars": False,  
        "bodies": True,  
        ...  
    }  
}
```

4. Установите "gdpr": True, чтобы включить размытие на видеостене всех объектов без совпадений.

Совет: Для того чтобы в полном объеме выполнить требования законов о защите персональных данных, руководствуйтесь *данной инструкцией*.

```
FFSECURITY_UI_CONFIG = {  
    ...  
    "available_video_wall_features": {  
        ...  
        "gdpr": True  
    }  
}
```

5. Перезапустите findface-security.

```
sudo systemctl restart findface-security.service
```

См.также:

Включение защиты персональных данных

Руководство по интеграции

Данная глава посвящена возможностям интеграции с FindFace Multi. Для интеграции своей системы используйте HTTP API и веб-хуки. Также обратите внимание на список наших готовых интеграций с партнерами.

4.1 HTTP API

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*.

4.2 Вебхуки

Вы можете настроить FindFace Multi для автоматической отправки уведомлений об определенных событиях, эпизодах, записях счетчиков и активациях областей на заданный URL-адрес. Для этого создайте и настройте вебхук. При наступлении нужного события/эпизода/записи счетчика/активации области 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)*

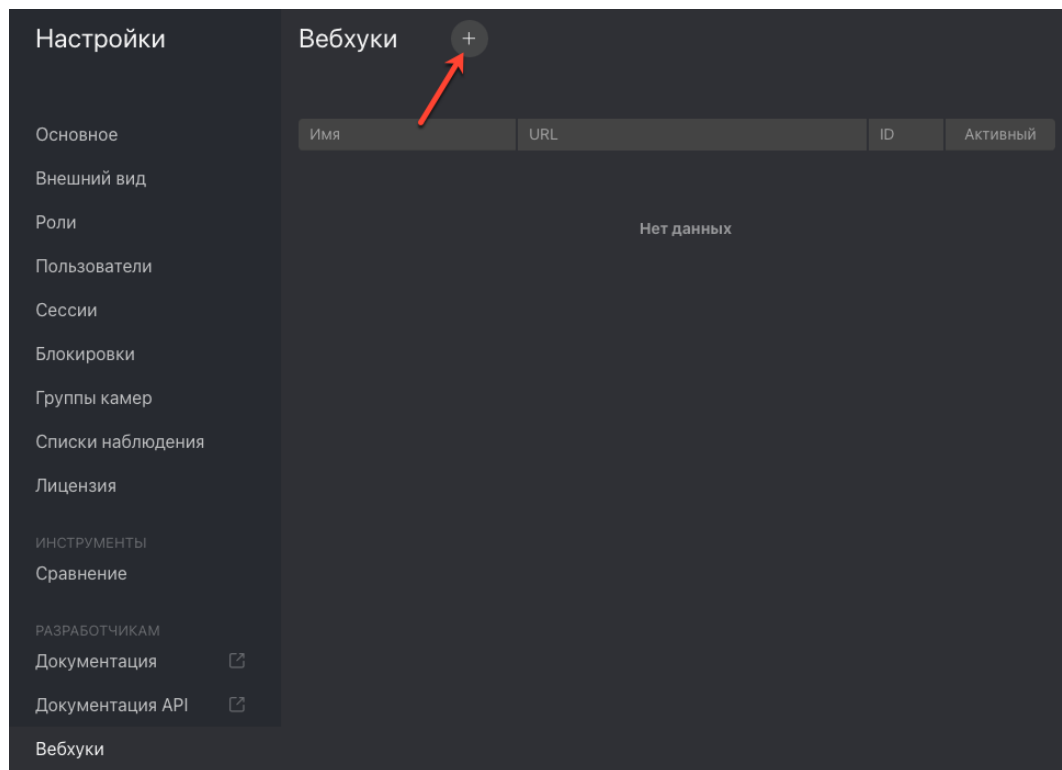
4.2.1 Настройка вебхука

Важно: Для создания вебхука необходимы права администратора.

Примечание: Для того чтобы использовать вебхуки, обязательно укажите по крайней мере один из параметров `SERVICE_EXTERNAL_ADDRESS/EXTERNAL_ADDRESS` в файле `/etc/findface-security/config.ru`.

Для создания вебхука выполните следующие действия:

1. Перейдите на вкладку *Настройки*. Выберите *Вебхуки*.
2. Нажмите **+**.



3. Введите имя вебхука.

Создать вебхук

• Заголовок вебхука

Контроль доступа

• URL

<http://ntechlab.com/webhooks/access/gate4>

Количество сообщений в пакете

1

Количество попыток отправки (0 - не ограничено)

10

Чтобы не получать старые события в случае временного отключения получателя, необходимо установить значение параметра в 1.

Фильтры

```
{
  "face_events": {
    "allowed_bs_types": [
      "overall",
      "realtime"
    ],

```

☒ Активный

[Сохранить](#) [Назад](#)

5.2.9

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, записях счетчика и активациях области, удовлетворяющих заданным фильтрам. Фильтровать события можно по следующим параметрам:

События распознавания (лицо, силуэт, автомобиль):

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.
- `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.
- `confidence_gte`: минимальное значение уверенности алгоритма, 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 лиц 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 силуэтов 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 автомобилями:

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 автомобилей в эпизоде с картотекой, логическое значение. Установите `true`, если только эпизоды с совпадениями автомобилей, или `false`, если только эпизоды без совпадений автомобилей должны вызывать отправку оповещения.

- `events_count_gte`: минимальное количество событий в эпизоде, число.
- `events_count_lte`: максимальное количество событий в эпизоде, число.

Записи 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 записи счетчика, число.
- `faces_lte`: максимальное количество лиц в записи счетчика, число.
- `silhouettes_gte`: минимальное количество силуэтов в записи счетчика, число.
- `silhouettes_lte`: максимальное количество силуэтов в записи счетчика, число.
- `cars_gte`: минимальное количество автомобилей в записи счетчика, число.
- `cars_lte`: максимальное количество лиц в записи 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.

Активации областей:

Секция: `areas`.

- `area_in`: id области, число.
- `camera_group_in`: id группы камер, число.
- `camera_in`: id камеры, число.

```
{
    "face_events": {
        "allowed_bs_types": [
            "overall",
            "realtime"
        ],
        "camera_group_in": [],
    },
}
```

(continues on next page)

(продолжение с предыдущей страницы)

```

        "camera_in": [],
        "matched_lists_in": [],
        "matched_card_in": [],
        "matched": true,
        "confidence_gte": 0.75
    },
    "body_events": {
        "allowed_bs_types": [
            "overall",
            "realtime"
        ],
        "camera_group_in": [],
        "camera_in": [],
        "matched_lists_in": [],
        "matched_card_in": [],
        "matched": true,
        "confidence_gte": 0.75
    },
    "car_events": {
        "allowed_bs_types": [
            "overall",
            "realtime"
        ],
        "camera_group_in": [],
        "camera_in": [],
        "matched_lists_in": [],
        "matched_card_in": [],
        "matched": true,
        "confidence_gte": 0.75
    },
    "human_episodes": {
        "allowed_types": [
            "episode_open",
            "episode_event",
            "episode_close"
        ],
        "camera_group_in": [],
        "camera_in": [],
        "matched_lists_in": [],
        "face_matched": true,
        "body_matched": true,
        "events_count_gte": 0,
        "events_count_lte": 999
    },
    "car_episodes": {
        "allowed_types": [
            "episode_open",
            "episode_event",
            "episode_close"
        ],
        "camera_group_in": [],
        "camera_in": [],

```

(continues on next page)

(продолжение с предыдущей страницы)

```

        "matched_lists_in": [],
        "car_matched": true,
        "events_count_gte": 0,
        "events_count_lte": 999
    },
    "counters": {
        "counter_in": [],
        "camera_group_in": [],
        "camera_in": [],
        "faces_gte": 1,
        "faces_lte": 100,
        "silhouettes_gte": 1,
        "silhouettes_lte": 100,
        "cars_gte": 1,
        "cars_lte": 100,
        "proximity_min_lte": 100,
        "proximity_min_gte": 0,
        "proximity_avg_lte": 100,
        "proximity_avg_gte": 0,
        "proximity_max_lte": 100,
        "proximity_max_gte": 0
    },
    "areas": {
        "area_in": [],
        "camera_group_in": [],
        "camera_in": []
    }
}

```

Важно: Используйте только фильтры, соответствующие цели поиска. Для исключения 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 автомобилями:

```
{ "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 *Активный*.
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, записях счетчиков и активациях областей. Оповещения отправляются в следующих форматах:

- Событие распознавания лица
- Событие распознавания силуэта
- Событие распознавания автомобиля
- Открытие эпизода с человеком

- Открытие эпизода с автомобилем
- Заккрытие эпизода с человеком
- Заккрытие эпизода с автомобилем
- Запись счетчика
- Активация области

Для просмотра статуса отправки вебхука в FindFace Multi выполните следующую команду:

```
sudo journalctl -u findface-security.service | grep 'Webhook'
```

Ответ в случае успеха:

```
May 30 14:13:43 ffsecurity[12441]: INFO      [Webhook(id=6) worker(type=face_events]
↳<queue: 0> Sent batch(len=1, type="face_events"): ['4355024961160384430']
May 30 14:13:43 ffsecurity[12441]: INFO      [SC:OQsrsPV9] [Webhooks manager-38bc5]
↳Processing message(type="face_events:event_created"). Consumer reception delta: 0.
↳003450
May 30 14:13:43 ffsecurity[12441]: INFO      [Webhook(id=6) worker(type=face_events]
↳<queue: 0> Sent batch(len=1, type="face_events"): ['4355024961658847580']
May 30 14:13:44 ffsecurity[12441]: INFO      [SC:JtRz2Vuo] [Webhooks manager-38bc5]
↳Processing message(type="face_events:event_created"). Consumer reception delta: 0.
↳001263
May 30 14:13:44 ffsecurity[12441]: INFO      [Webhook(id=6) worker(type=face_events]
↳<queue: 0> Sent batch(len=1, type="face_events"): ['4355024962087522421']
May 30 14:13:44 ffsecurity[12441]: INFO      [SC:9AnzRJwU] [Webhooks manager-38bc5]
↳Processing message(type="face_events:event_created"). Consumer reception delta: 0.
↳001691
May 30 14:13:44 ffsecurity[12441]: INFO      [Webhook(id=6) worker(type=face_events]
↳<queue: 0> Sent batch(len=1, type="face_events"): ['4355024962355957878']
```

Ответ, если попытка была неудачной:

```
May 30 14:18:49 ffsecurity[12441]: INFO      [SC:sp34rVQR] [Webhooks manager-38bc5]
↳Processing message(type="face_events:event_created"). Consumer reception delta: 0.
↳001376
May 30 14:18:49 ffsecurity[12441]: WARNING [Webhook(id=6) worker(type=face_events]
↳<queue: 0> Error sending webhook: Cannot connect to host 127.0.0.1:8888 ssl:None
↳[Connection refused]. Attempt 1 out of 10. Next attempt in 0.270 seconds.
May 30 14:18:50 ffsecurity[12441]: WARNING [Webhook(id=6) worker(type=face_events]
↳<queue: 0> Error sending webhook: Cannot connect to host 127.0.0.1:8888 ssl:None
↳[Connection refused]. Attempt 2 out of 10. Next attempt in 0.729 seconds.
May 30 14:18:50 ffsecurity[12441]: INFO      [SC:zUhLHNxN] [Webhooks manager-38bc5]
↳Processing message(type="face_events:event_created"). Consumer reception delta: 0.
↳001368
May 30 14:18:50 ffsecurity[12441]: INFO      [SC:1Q66tcUS] [Webhooks manager-38bc5]
↳Processing message(type="face_events:event_created"). Consumer reception delta: 0.
↳001386
May 30 14:18:50 ffsecurity[12441]: WARNING [Webhook(id=6) worker(type=face_events]
↳<queue: 2> Error sending webhook: Cannot connect to host 127.0.0.1:8888 ssl:None
```

(continues on next page)

(продолжение с предыдущей страницы)

```

↪ [Connection refused]. Attempt 3 out of 10. Next attempt in 1.968 seconds.
May 30 14:18:52 ffsecurity[12441]: WARNING [Webhook(id=6) worker(type=face_events)
↪ <queue: 2> Error sending webhook: Cannot connect to host 127.0.0.1:8888 ssl:None
↪ [Connection refused]. Attempt 4 out of 10. Next attempt in 5.314 seconds.
May 30 14:18:55 ffsecurity[12441]: INFO [SC:5kl6zGrF] [Webhooks manager-38bc5]
↪ Processing message(type="face_events:event_created"). Consumer reception delta: 0.
↪ 001542
May 30 14:18:58 ffsecurity[12441]: WARNING [Webhook(id=6) worker(type=face_events)
↪ <queue: 3> Error sending webhook: Cannot connect to host 127.0.0.1:8888 ssl:None
↪ [Connection refused]. Attempt 5 out of 10. Next attempt in 14.349 seconds.

```

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 `/etc/findface-security/config.py` и установите в нем `'VERBOSE_WEBHOOKS': True`:

```

sudo vi /etc/findface-security/config.py

...
FFSECURITY = {
    ...
    # send serialized cards, card-lists, camera and camera groups in webhooks
    'VERBOSE_WEBHOOKS': True,
    ...
}
...

```

После внесения изменений обязательно перезапустите сервис `findface-security`.

```
sudo systemctl restart findface-security.service
```

В подробном режиме оповещения по вебхуку имеют следующие форматы:

- Событие распознавание лица (verbose)
- Событие распознавание силуэта (verbose)
- Событие распознавание автомобиля (verbose)
- Открытие эпизода с человеком (verbose)
- Открытие эпизода с автомобилем (verbose)
- Закрытие эпизода с человеком (verbose)
- Закрытие эпизода с автомобилем (verbose)
- Запись счетчика (verbose)
- Активация области (verbose)

4.3 Интеграции с партнерами

4.3.1 Genetec Security Center

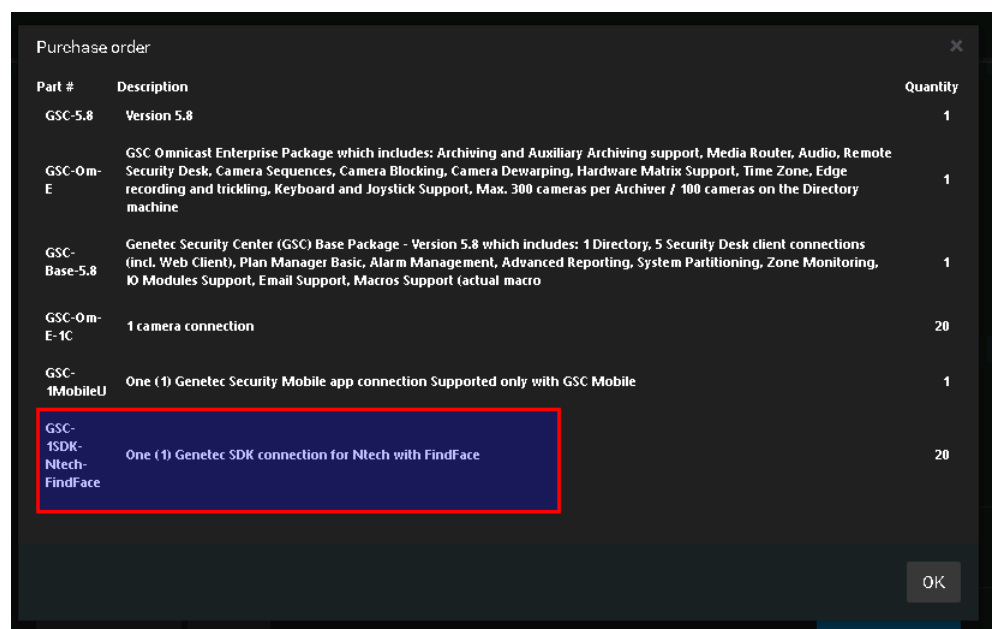
Интеграция FindFace Multi с программным комплексом Genetec Security Center позволяет добавлять функционал распознавания лиц в системы безопасности на базе Genetec.

Настройка интеграции

Интеграция с Genetec Security Center реализуется через плагин `findface-genetec`. По умолчанию плагин отключен.

Перед настройкой интеграции на стороне FindFace Multi разверните программное обеспечение Genetec Web SDK и Media Gateway и создайте в Genetec Security Center оповещение **Alarm**, которое будет отображаться при наступлении в FindFace Multi события распознавания лица.

Важно: Для того чтобы интеграция Genetec-FindFace работала, вам также понадобится приобрести соответствующую лицензию у Genetec (номер по каталогу лицензий `GSC-1SDK-Ntech-FindFace`) и активировать ее в Genetec Security Center.



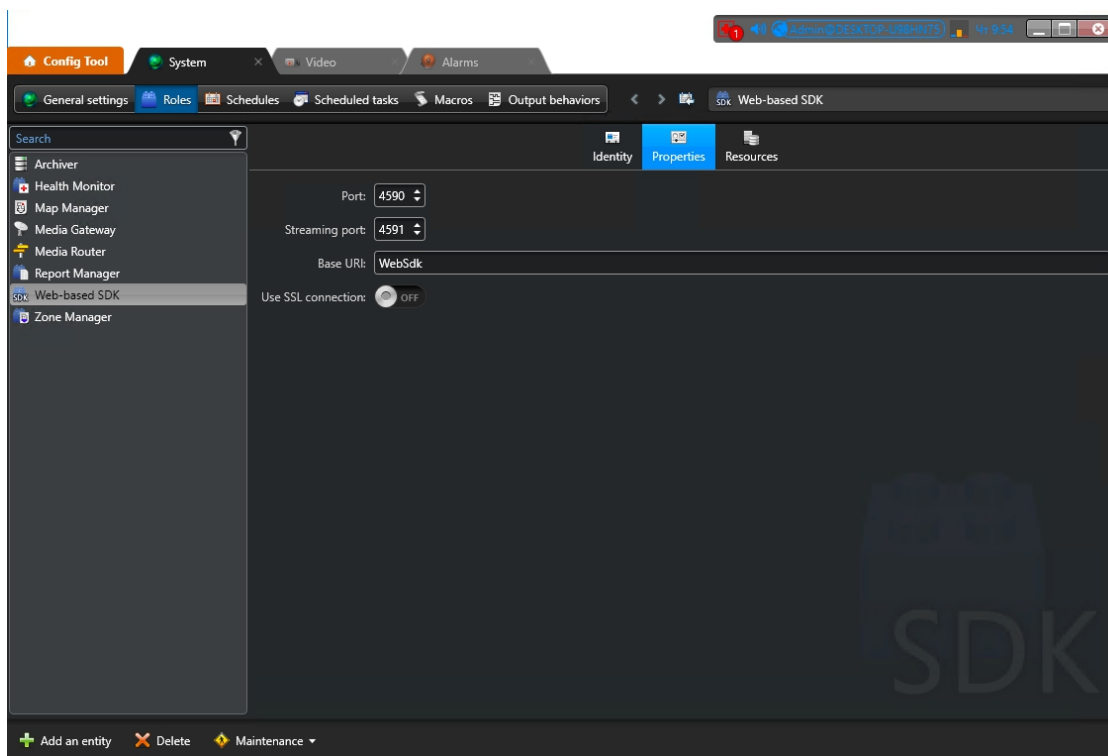
В этой главе:

- Настройка Genetec Web SDK и Media Gateway
- Создание оповещения в Genetec Security Center
- Включение интеграции Genetec в FindFace Multi
- Настройка точек доступа в FindFace Multi
- Импорт камер из Genetec Security Center

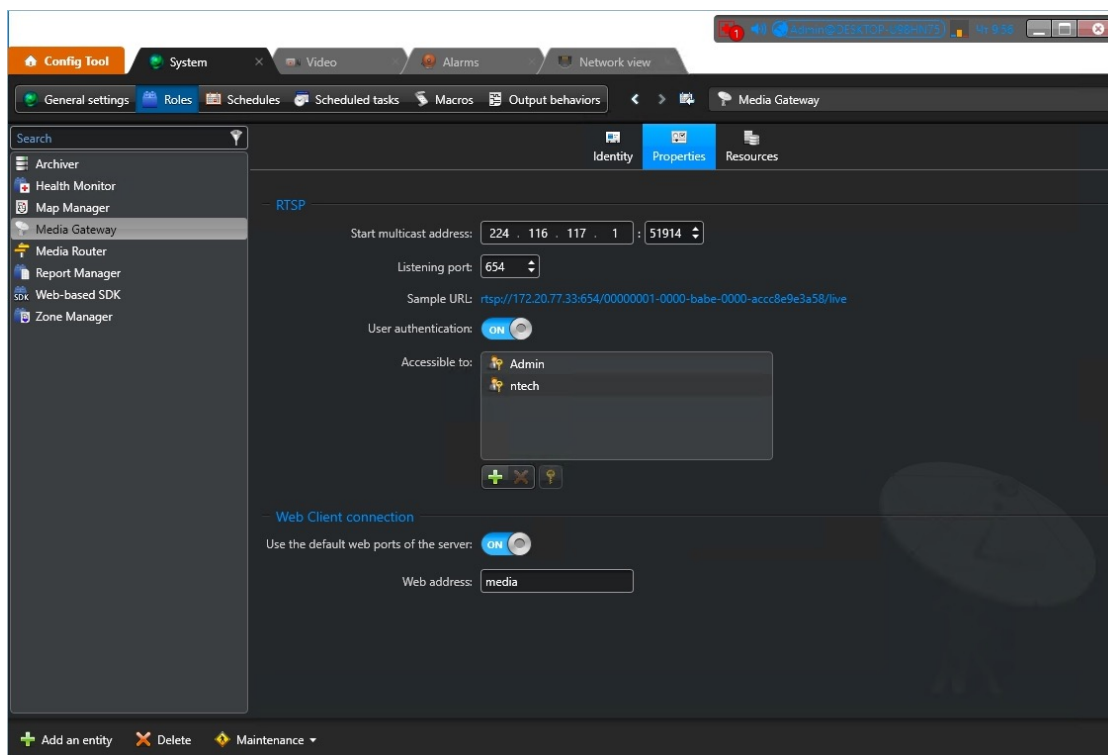
- *Создание списков наблюдения и карточек в FindFace Multi*

Настройка Genetec Web SDK и Media Gateway

Для того чтобы развернуть Web SDK, используйте ПО Genetec Config Tool. Детали настройки приведены в официальной справочной документации *Security Center Administrator Guide* -> *Chapter 52: Role Types* -> *Web-based SDK configuration tabs*.



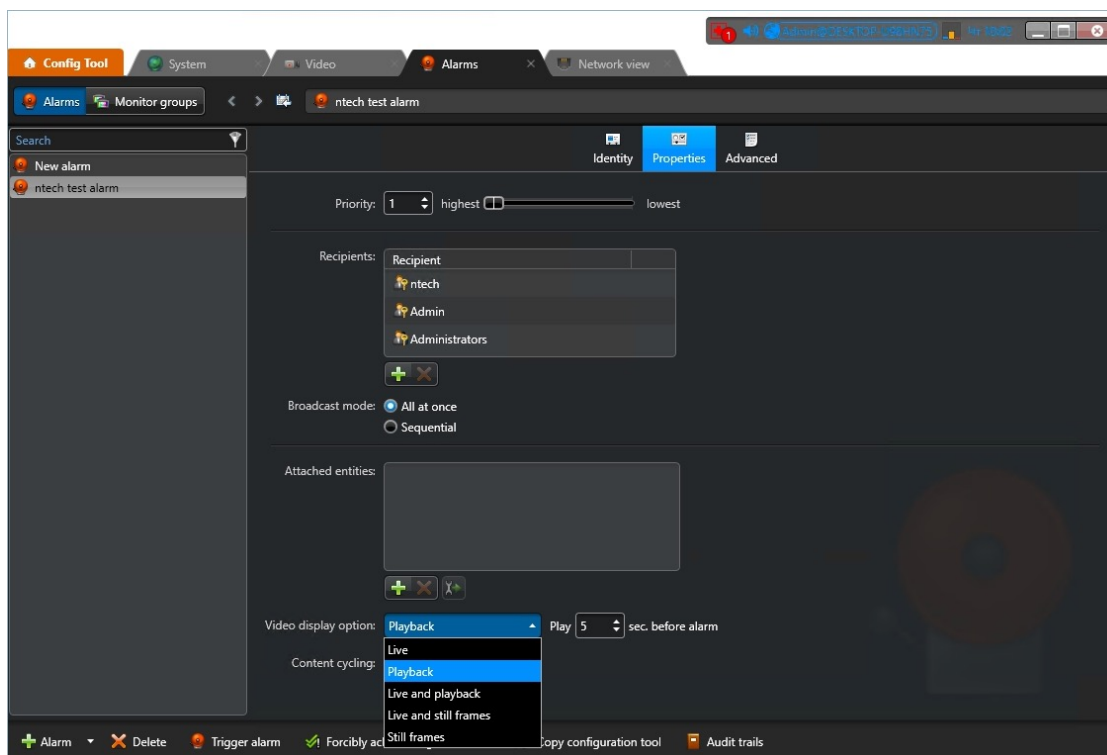
Для того чтобы развернуть Media Gateway в Genetec Config Tool, ознакомьтесь с содержанием главы *Security Center Administrator Guide* -> *Chapter 24: Video Deployment*.



Важно: Убедитесь, что фаервол настроен таким образом, что порты WebSDK и Media Gateway остаются открытыми.

Создание оповещения в Genetec Security Center

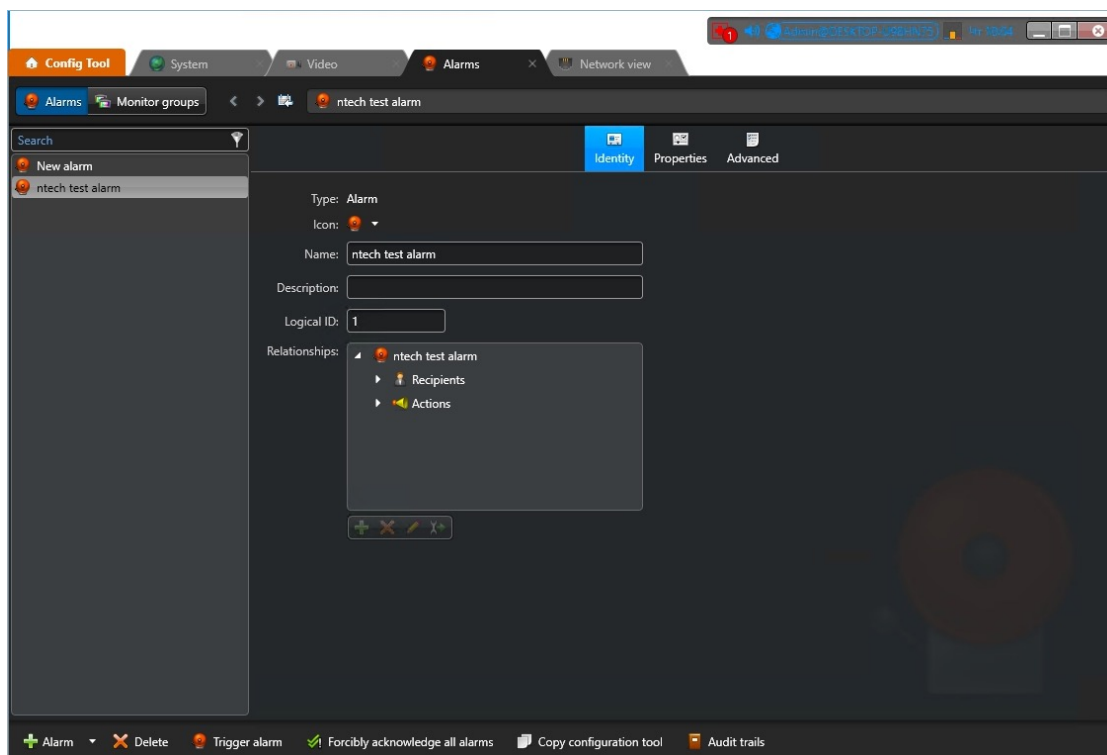
Создайте и настройте новое оповещение **Alarm** в Genetec Config Tool, руководствуясь разделом *Security Center Administrator Guide -> Chapter 48: Alarms -> Creating Alarms*.



Совет: На вкладке *Properties* выберите ту опцию отображения видео *Video display option*, которая в наибольшей степени соответствует вашим needs. Доступные опции *Live*, *Playback*, и т. д.

Совет: Для того чтобы активировать операции с оповещением Alarm Procedures и автоповорот видео непосредственно во всплывающем окне оповещения, включите *Content cycling*.

При настройке интеграции на стороне FindFace Multi вам потребуется ввести логическое id оповещения, которое задается на вкладке *Identity*.



Включение интеграции Genetec в FindFace Multi

Для того чтобы активировать интеграцию Genetec в FindFace Multi, выполните следующие действия:

1. Включите плагин `findface-genetec`. Для этого откройте файл конфигурации `/etc/findface-security/config.py` и раскомментируйте строку `INSTALLED_APPS.append('ffsecurity_genetec')`. Убедитесь, что по крайней мере один из следующих параметров задан: `SERVICE_EXTERNAL_ADDRESS` или `EXTERNAL_ADDRESS`.

```
sudo vi /etc/findface-security/config.py

...
# SERVICE_EXTERNAL_ADDRESS is prioritized for FFSecurity webhooks and Genetec
# plugin.
# EXTERNAL_ADDRESS is used instead if SERVICE_EXTERNAL_ADDRESS is not provided.
# You must provide either SERVICE_EXTERNAL_ADDRESS or EXTERNAL_ADDRESS in order
# to be able to work with FFSecurity webhooks and Genetec plugin.
SERVICE_EXTERNAL_ADDRESS = 'http://127.0.0.1'
# EXTERNAL_ADDRESS is used to access objects created inside FFSecurity via external
# links.
EXTERNAL_ADDRESS = ''

...
# FINDFACE SECURITY PLUGINS
# =====
# Uncomment lines below to enable plugins. Please consult documentation for
# a plugin specific settings.
...
```

(continues on next page)

(продолжение с предыдущей страницы)

```
# ===== Genetec =====  
INSTALLED_APPS.append('ffsecurity_genetec')
```

2. Перенесите схему базы данных из FindFace Multi в PostgreSQL и заново создайте группы пользователей с *предустановленными* правами.

```
sudo findface-security migrate  
sudo findface-security create_groups
```

3. Перезапустите findface-security.

```
sudo systemctl restart findface-security.service
```

Настройка точек доступа в FindFace Multi

Для того чтобы установить соединение между FindFace Multi и Genetec Security Center, выполните следующие действия:

1. Перейдите на вкладку *Настройки*. Выберите *Genetec*.

Настройки genetec
Состояние: -

Конфигурация Камеры

Сервер

Host
127.0.0.1

Port
4590

Пользователь
admin

Пароль

Base uri
WebSDK

Media

Media host
127.0.0.1

Media port
654

Пользователь
admin

Пароль

2. В секциях *Сервер* и *Media*, укажите *настройки* точек доступа Web SDK и Media Gateway.

Важно: Порты WebSDK и Media Gateway должны быть открыты.

- В секции `guilabel:Ids` укажите *логический id* оповещения **Alarm**, которое будет отображаться в Genetec Security Center при наступлении события распознавания лица в FindFace Multi.

- Нажмите *Сохранить*. Если соединение с Genetec Security Center успешно установлено, статус будет автоматически изменен на *Сконфигурирован*.

Импорт камер из Genetec Security Center

Как только соединение с Genetec Security Center установлено, можно импортировать камеры. Для этого выберите *Камеры* на вкладке *Genetec* и нажмите *Импорт*.

Данное действие создаст *группу камер Genetec*, включающую в себя все камеры из Genetec Security Center.

Id	Name	Active
1	1	<input checked="" type="checkbox"/>
2	Genetec Imported from Genetec Security Center	<input checked="" type="checkbox"/>

Для того чтобы посмотреть список камер, на панели навигации FindFace Multi перейдите на вкладку *Камеры*. Для того чтобы исключить камеру из распознавания лиц, просто деактивируйте ее в этом списке.

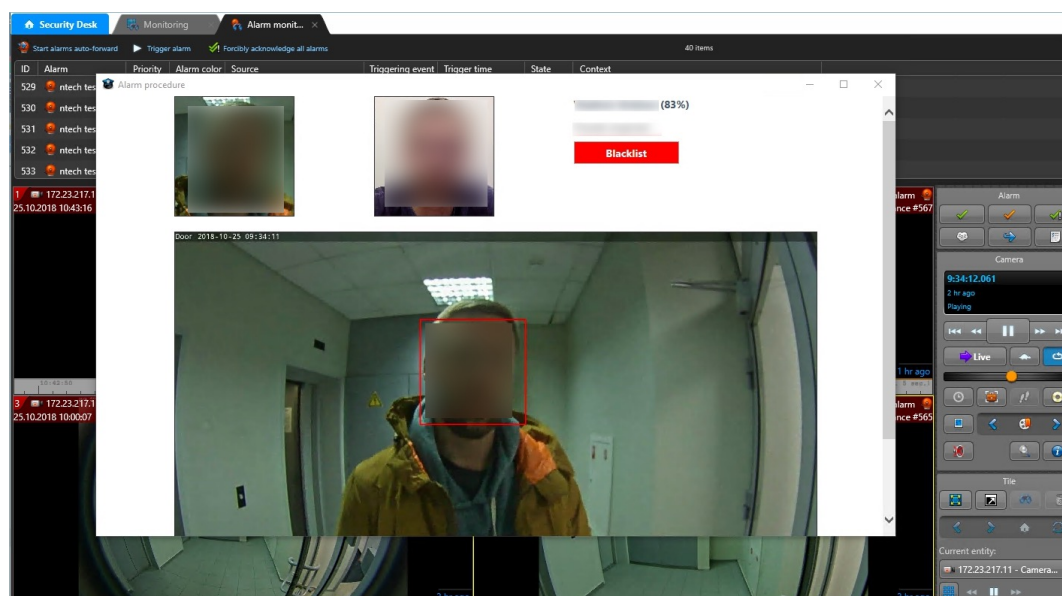
Создание списков наблюдения и карточек в FindFace Multi

После настройки точек доступа и импорта камер завершите интеграцию, создав *картотеку*. После этого оповещения о событиях распознавания лиц будут автоматически отправляться в Genetec Security Center. См. *Оповещения в Genetec Security Center*.

Оповещения в Genetec Security Center

Каждое событие распознавания лица с камеры Genetec, для которого найдена карточка, активирует соответствующее оповещение **alarm** в Genetec Security Center. Каждое оповещение, отправленное FindFace Multi, связывается с камерой-источником события распознавания лица, поэтому вы можете сразу же просматривать живое или архивное видео в задаче Alarm Monitoring в Genetec Security Desk. FindFace Multi также использует операции с оповещением Alarm Procedures для обеспечения пользователя дополнительными данными по событию, такими как:

- обнаруженное на видео лицо
- найденное похожее лицо из картотеки
- имя человека и комментарий из картотеки
- степень схожести лиц (уверенность алгоритма в совпадении)
- название списка наблюдения
- полный кадр



Обработка полученного оповещения о распознавания лица выполняется аналогично другим оповещениям в Genetec Security Center.

4.3.2 Axxon Next

Интеграция FindFace Multi с программным комплексом Axxon Next позволяет обрабатывать видеопотоки из системы безопасности на базе Аххон и анализировать их на предмет наличия объектов из картотеки.

Интеграция с Аххон Next выполняется с использованием плагина `ffsecurity_axxon`.

Для того чтобы настроить интеграцию с Аххон Next в ОС Ubuntu, выполните следующие действия:

1. Активируйте плагин, раскомментировав в файле конфигурации `/etc/findface-security/config.py` строку `INSTALLED_APPS.append('ffsecurity_axxon')`.

```
sudo vi /etc/findface-security/config.py

...

# =====
# FINDFACE SECURITY PLUGINS
# =====
# Uncomment lines below to enable plugins. Please consult documentation for
# a plugin specific settings.

# ===== Axxon =====
INSTALLED_APPS.append('ffsecurity_axxon')
```

2. В файле конфигурации раскомментируйте секцию `AXXON`. Заполните ее по аналогии с примером ниже. В параметре `api` укажите IP-адрес сервера и API-порт Аххон Next, по которому FindFace Multi будет обращаться к API Аххон и за HLS-потоками архива. В параметре `rtsp` укажите общий сегмент в адресах видеопотоков Аххон. `name`, `user`, `password`: имя сервера Аххон Next и учетные данные для доступа к нему.

```
===== Axxon =====
INSTALLED_APPS.append('ffsecurity_axxon')

AXXON = [
    {
        'name': 'my_axxon',
        'api': 'http://192.168.0.2:8000',
        'rtsp': 'rtsp://192.168.0.2:50554/',
        'user': 'cinderella',
        'password': '1@love$ocean8',
    }
]
```

Примечание: Вы можете установить соединение с несколькими серверами Аххон Next, размножив содержимое секции `AXXON` следующим образом:

```
AXXON = [
    {
        'name': 'my_axxon1',
        'api': 'http://192.168.0.2:8000',
        'rtsp': 'rtsp://192.168.0.2:50554/',
        'user': 'user1',
```

(continues on next page)

(продолжение с предыдущей страницы)

```

    'password': 'password1',
  },
  {
    'name': 'my_axxon2',
    'api': 'http://192.168.0.3:8000',
    'rtsp': 'rtsp://192.168.0.3:50554/',
    'user': 'user2',
    'password': 'password2',
  },
  {
    'name': 'my_axxon3',
    'api': 'http://192.168.0.4:8000',
    'rtsp': 'rtsp://192.168.0.4:50554/',
    'user': 'user3',
    'password': 'password3',
  }
]

```

3. Создайте представления камер Axxon Next в FindFace Multi (см. *Управление видеокameraми. Видеомагнитофон*). URL-адрес в представлении камер должен быть указан в формате `<server>:<friendlyNameLong>`, где `server` – имя сервера Axxon Next, указанное на предыдущем шаге в параметре `name`, `friendlyNameLong` – имя камеры на сервере Axxon Next. Узнать имя камеры можно в пользовательском интерфейсе Axxon или через Axxon API с помощью следующей команды:

```

curl http://user:password@127.0.0.1/video-origins/

{
  "OLOLOE-DEV/DeviceIpint.vhod_1/SourceEndpoint.video:0:0" : {
    "friendlyNameLong" : "vhod_1.Vhod_1",
    "friendlyNameShort" : "Vhod_1",
    "origin" : "OLOLOE-DEV/DeviceIpint.vhod_1/SourceEndpoint.video:0:0",
    "state" : "signal_restored"
  }
}

```

Для камеры из нашего примера URL будет задан как `my_axxon:vhod_1.Vhod_1`.

На этом настройка интеграции будет завершена. Если интеграция настроена корректно, FindFace Multi будет выполнять проверку наличия объектов из картотеки в видеопотоках Axxon Next, а в событиях распознавания будут отображаться клипы видео из Axxon Next (при соответствующих настройках).

4.3.3 Sigur

Интеграция FindFace Multi со СКУД Sigur обеспечивает эффективный контроль управления доступом с использованием системы распознавания лиц.

В этом разделе:

- Особенности интеграции FindFace Multi и Sigur
- Установка и настройка FindFace Multi
- Установка и настройка ПО Sigur
 - Установка ПО Sigur
 - Настройка удаленного подключения к базе данных (опционально)
 - Настройка Сервера Sigur
 - Настройка Клиента Sigur
 - Добавление пользователей
- Проверка синхронизации систем

Особенности интеграции FindFace Multi и Sigur

- Интеграция не поддерживает пагинацию.

Клиент Sigur отдает FindFace Multi все лица сразу в одном json. Для большого количества лиц это занимает много времени (10к лиц более 30 секунд). При настройке необходимо увеличение таймаутов, а также использование версии сервера/клиента Sigur с увеличенными таймаутами ($\geq 1.0.60.23$).

- Интеграция не поддерживает обработку ошибок фотографий.

В случае если на фото более одного лица либо качество лица ниже требуемого уровня, досье будет создаваться без фото. При наличии на фото нескольких лиц рекомендуется использовать *опцию biggest* для выбора самого большого лица.

Установка и настройка FindFace Multi

Установка FindFace Multi не отличается от *типовой*.

Важно: Если вы настраиваете интеграцию с уже существующей системой FindFace Multi, убедитесь, что в ней в списке наблюдения `Default watch list` с `id = 1` отсутствуют лица, поскольку по умолчанию синхронизация досье Sigur идет с данным списком наблюдения и он должен быть пустым.

Для включения интеграции с Sigur выполните следующие действия:

1. Откройте файл конфигурации `/etc/findface-security/config.py` и раскомментируйте блок плагина SIGUR и блок CUSTOM_FIELDS, как показано в примере ниже.

```

sudo vi /etc/findface-security/config.py

#===== Sigur =====
# keep in mind, that SIGUR plugin also uses CUSTOM_FIELDS and THUMBNAIL_MAX_WIDTH
↳ settings
INSTALLED_APPS.append('ffsecurity_sigur')
SIGUR = {
    'LOGIN': 'admin',
    'PASSWORD': 'admin',
    'MF_SELECTOR': 'biggest', # what to do with several faces in sigur person
↳ photo; allowed ['biggest', 'reject']
    'ONLY_RT_EVENTS': True, # only events with bs_type == realtime,
    'EVENT_DELAY': 0.004 # minimum time between two events of same person in
↳ seconds.
    # If interval between two events with same person is less, than this value,
↳ second event will be dropped
}

...
FFSECURITY = {

    # -- Custom model fields --
    # Edit CUSTOM_FIELDS -> `human_card` section to customize human card 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
                },
            ]
        }
    }
}

```

(continues on next page)

(продолжение с предыдущей страницы)

```

        {
            '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
# '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
#         }
#     ]
# }
}

```

2. Придумайте логин и пароль, которые будут использоваться для авторизации FindFace Multi в ПО Sigur, и укажите их в полях LOGIN и PASSWORD блока SIGUR.
3. В параметре MF_SELECTOR того же блока укажите, как должна выполняться обработка фото, если на нем присутствует несколько лиц: обработать самое большое лицо (**biggest**) или отклонить фотографию (**reject**).
4. Перенесите структуру основной базы данных из FindFace Multi в PostgreSQL. Для этого достаточно выполнить приведенную ниже команду.

```
sudo findface-security migrate
```

5. В веб-интерфейсе FindFace Multi откройте настройки той *камеры*, которую планируется использовать для распознавания лиц в интеграции с ПО Sigur. На вкладке *Аналитика* включите отправку

в ПО Sigur всех событий, установив `false` для параметра Буферный режим (`overall_only`).

6. Перезапустите сервис `findface-security`.

```
sudo systemctl restart findface-security.service
```

Установка и настройка ПО Sigur

Установка ПО Sigur

Для установки ПО Sigur выполните следующие действия:

1. На сайте Sigur узнайте имена deb-пакетов для самой актуальной версии Sigur.
2. Загрузите скрипт установки `sigur_install.sh` в произвольный каталог на сервере (например, в `/home/username/`). Замените в нем имена загружаемых пакетов на актуальные.

Пример скрипта

```
#!/bin/bash
#Очистка перед установкой
cd ~/
rm -R spnx/
mkdir spnx
cd spnx
#Для активации sudo
sudo ls
#Установка зависимостей
sudo apt update
sudo apt install -y openjdk-8-jre
sudo apt install -y mysql-server
#Загрузка пакетов - ИЗМЕНИТЕ ИМЕНА НА АКТУАЛЬНЫЕ
wget https://sigur.com/dl/spnxclient_1.1.1.5-0_all.deb
wget https://sigur.com/dl/spnxserver_1.1.1.5-0_amd64.deb
#Установка
sudo dpkg -i spnx*
#Запуск
sudo spnxadmin &
spnxclient &
```

3. Из каталога со скриптом сделайте скрипт исполняемым.

```
chmod +x sigur_install.sh
```

4. Выполните скрипт.

```
sudo ./sigur_install.sh
```

Настройка удаленного подключения к базе данных (опционально)

Для настройки базы данных выполните следующие действия:

1. Задайте пароль для пользователя **root**.

```
~/spnx$ sudo mysql

mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY
↪ 'YourRootPassword';
Query OK, 0 rows affected (0.00 sec)
```

2. В случае если требуется запускать клиент Sigur на машине из локальной сети, а не с **localhost**, использование пользователя **root** будет невозможным, поскольку для него есть только локальный доступ.

Для удаленного подключения к базе данных создайте отдельного пользователя и предоставьте ему права.

Ниже приведен пример создания пользователя **skud** с паролем **xxx**.

```
# mysql

MariaDB [(none)]> GRANT ALL PRIVILEGES ON `TC-DB-MAIN`.* TO 'skud' IDENTIFIED BY
↪ 'xxx';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON `TC-DB-LOG`.* TO 'skud';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)
```

Настройка Сервера Sigur

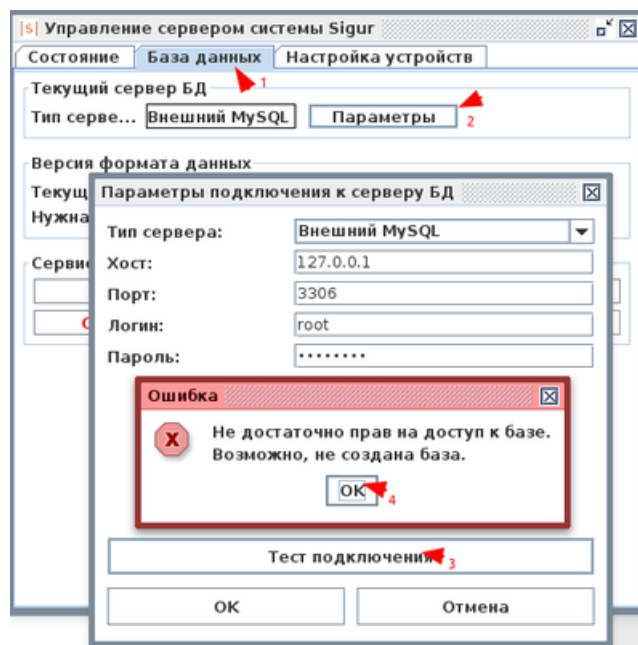
Для настройки и запуска Сервера Sigur выполните следующие действия:

1. Откройте панель администратора Сервера Sigur.

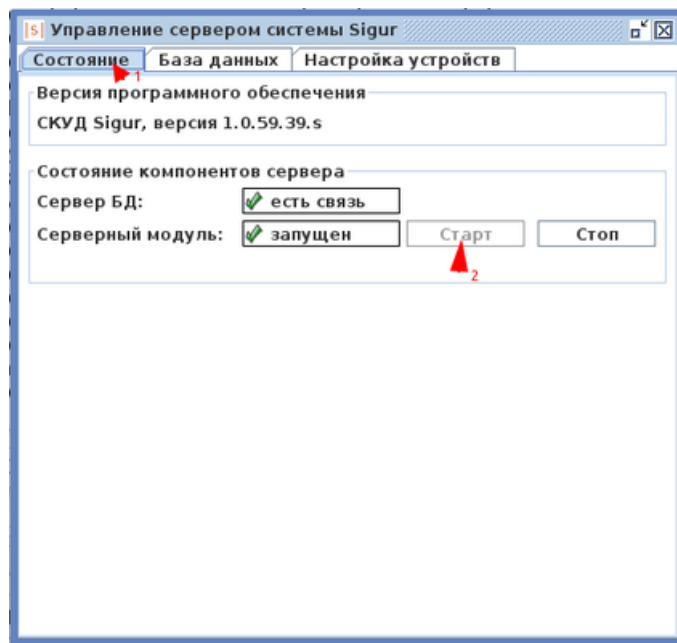
```
sudo spnxadmin
```

2. Создайте базу данных:

1. В открывшемся окне перейдите *База данных -> Параметры*. Введите логин и пароль от базы данных.
2. Выберите тип Сервера базы данных *Внешний MySQL*.
3. Выполните *Тест подключения*. Должна отобразиться ошибка **Недостаточно прав на доступ к базе...** Это значит подключение есть, но база данных еще не создана.
4. Нажмите *ОК*.



5. Для создания базы данных нажмите кнопку *Сброс/Создание базы*.
3. После создания базы данных запустите серверный модуль ПО Sigur. Для этого перейдите на вкладку *Состояние*. Нажмите кнопку *Старт* для пункта *Серверный модуль*.



Настройка Клиента Sigur

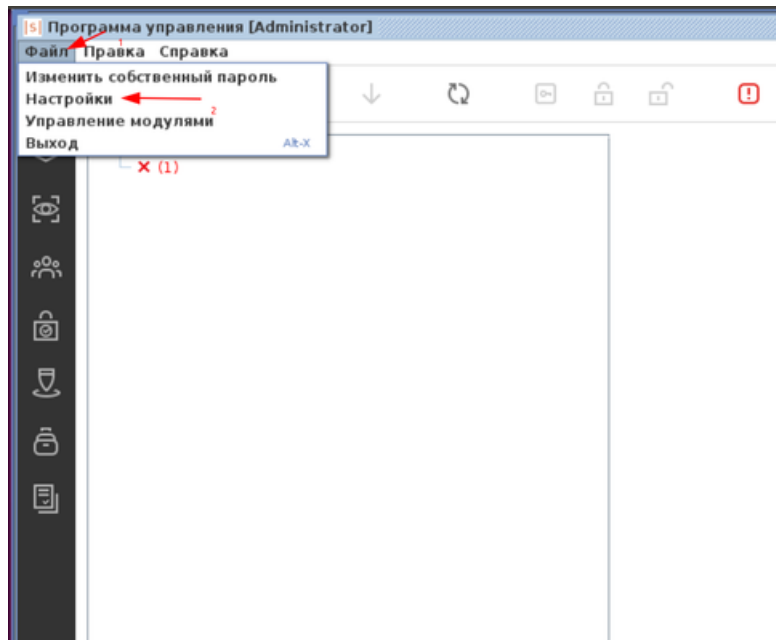
Для настройки Клиента Sigur выполните следующие действия:

1. Откройте панель клиента Sigur.

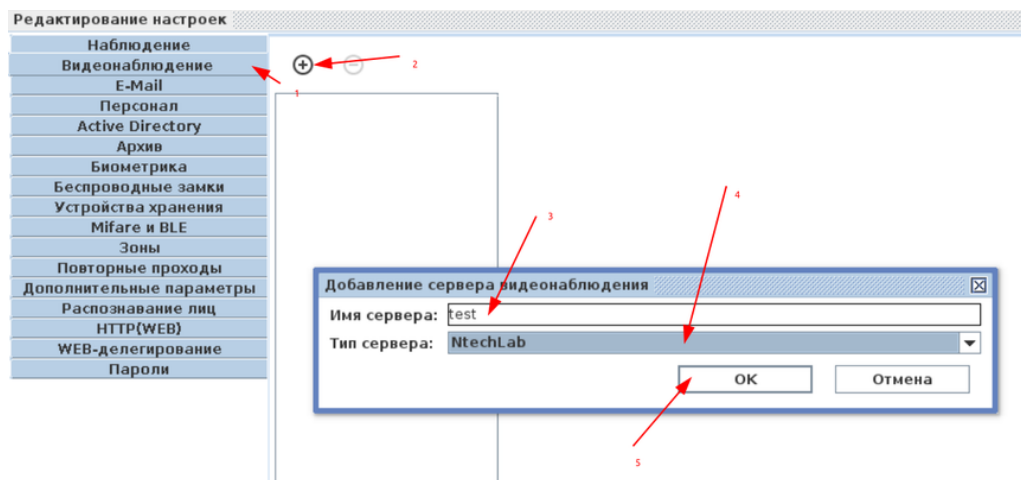
```
spxclient
```

2. Добавьте сервер FindFace Multi в Sigur:

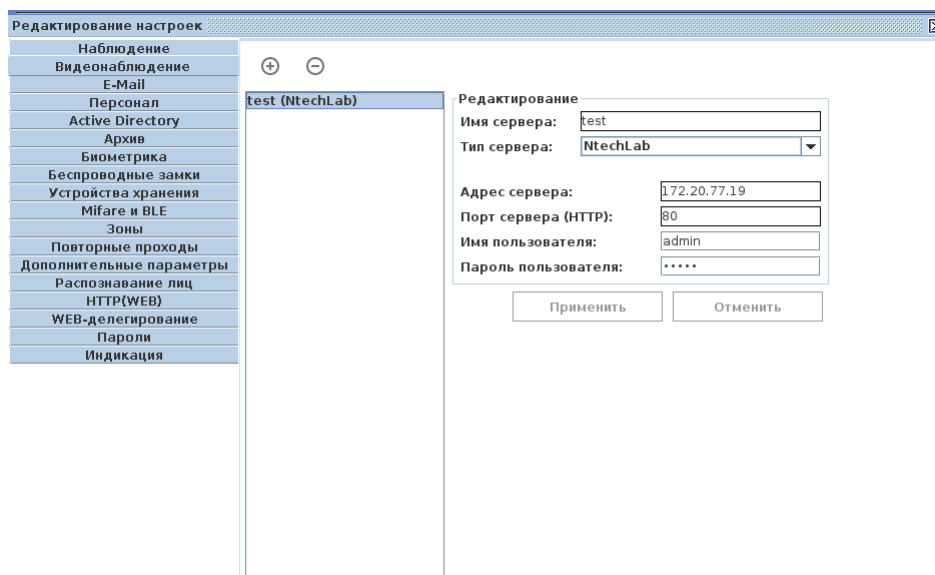
1. В главном меню перейдите *Файл -> Настройки -> Видеонаблюдение*.



2. В открывшемся окне выберите поставщика *NtechLab* и введите произвольное имя сервера.

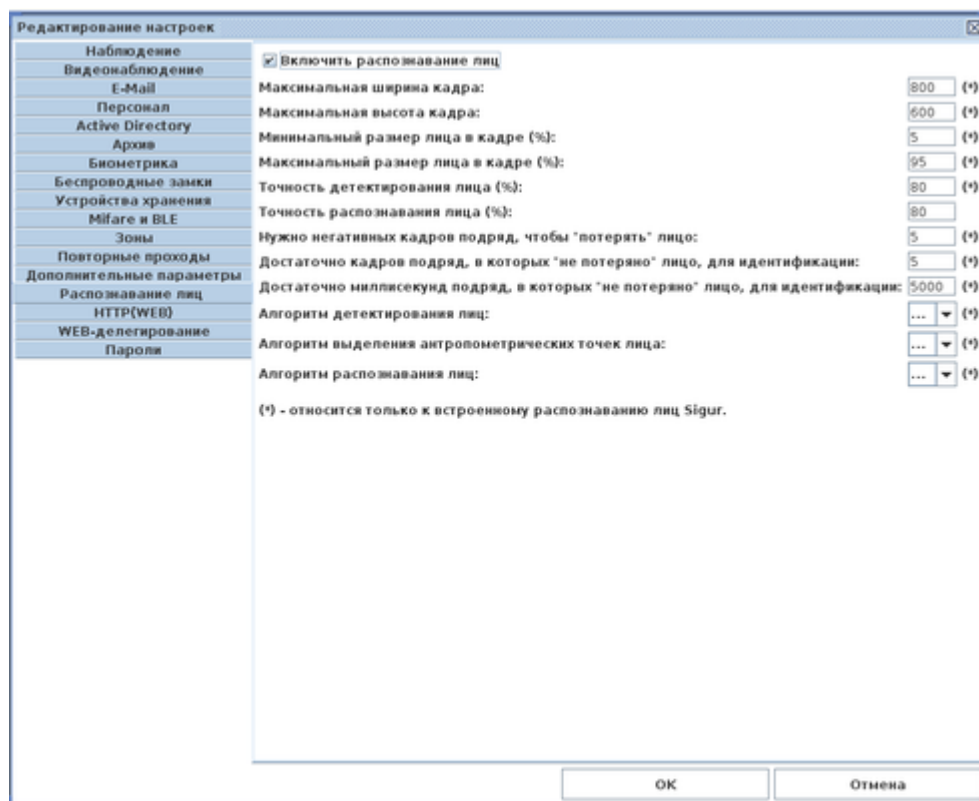


3. После того как в списке появится сервер, введите настройки доступа, которые вы указали в файле конфигурации *findface-security* (*логин и пароль* в блоке плагина Sigur). Порт для доступа Sigur к серверу FindFace Multi - 80.



4. Нажмите *Применить*.
3. Перейдите на вкладку *Распознавание лиц* и поставьте флажок *Включить распознавание лиц*.

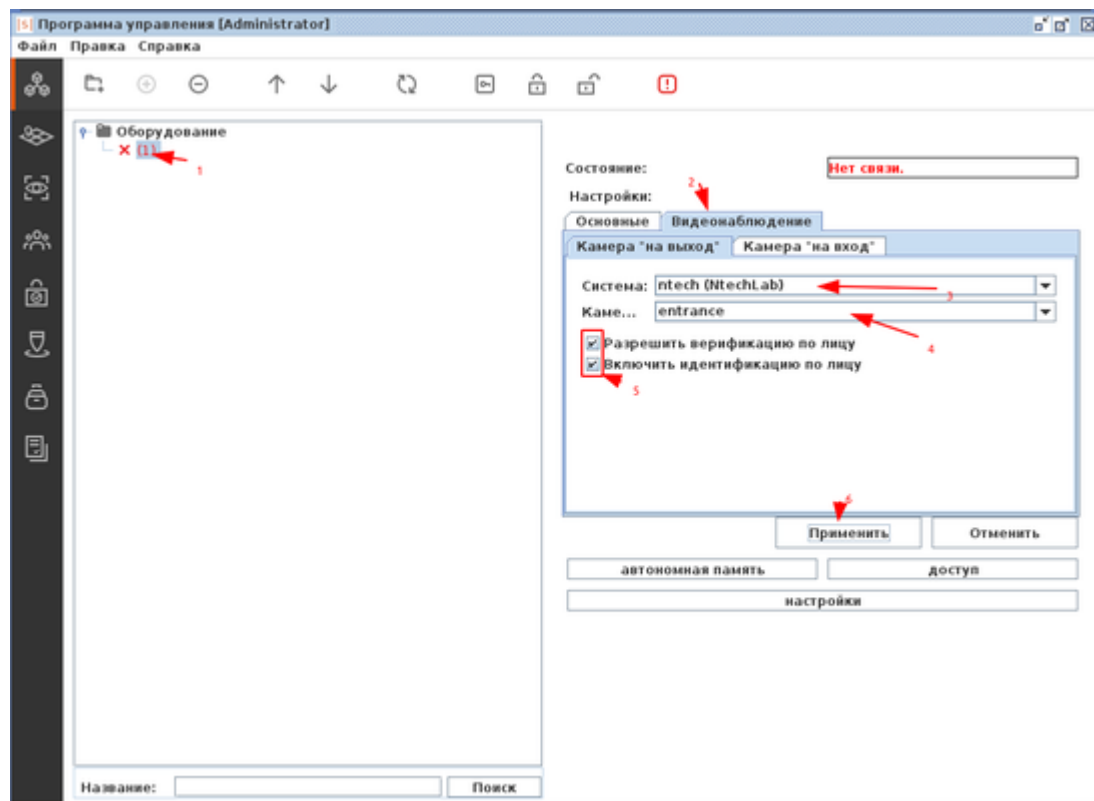
Важно: Значение в поле *Точность распознавания лица (%)* рекомендуется устанавливать ниже или равным установленному в *настройках FindFace Multi*.



4. Привяжите камеры к точкам входа:

Важно: Оборудование должно быть доступно в сети и заранее добавлено в Sigur.

1. Перейдите в главное окно программы, выберите вкладку *Оборудование*.
2. Выберите точку входа, перейдите на вкладку *Видеонаблюдение*.

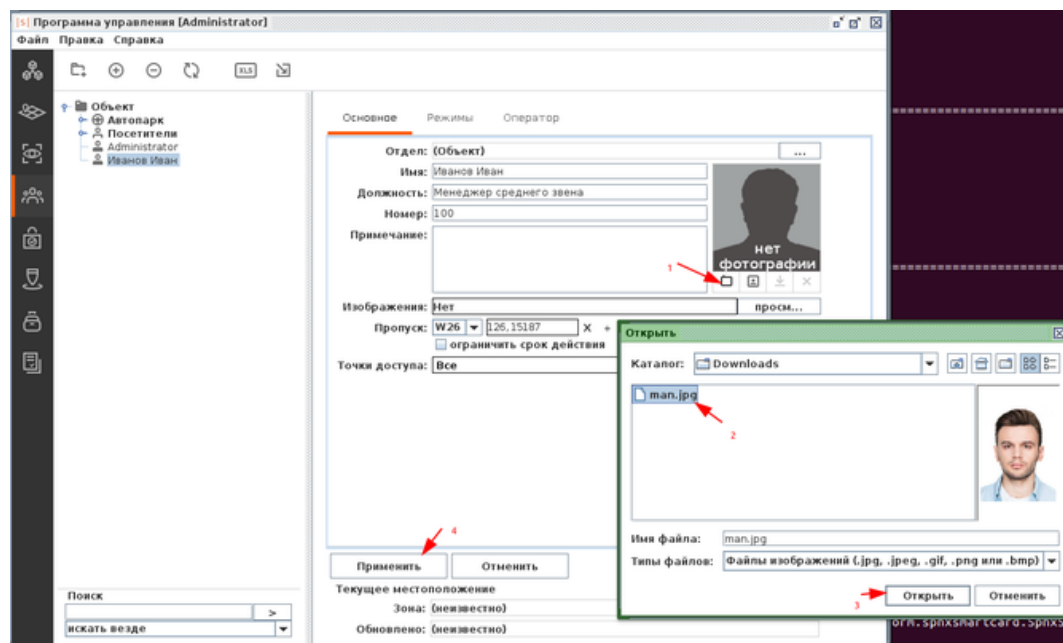


3. Установите следующие настройки:
 - Система: NtechLab.
 - Камера: в соответствии с камерой на данной точке входа.
 - флажки *Разрешить верификацию по лицу* и *Разрешить идентификацию по лицу* установлены.
4. Нажмите *Применить*.

Добавление пользователей

Для добавления пользователя выполните следующие действия:

1. В главном окне программы перейдите на вкладку *Персонал*. Нажмите кнопку *+* и в появившемся диалоговом окне введите имя пользователя. Нажмите *ОК*.



2. Прикрепите фотографию к профилю пользователя.

Важно: Если фотографии были сделаны в портретном режиме, необходимо их предварительно пересохранить в нужной ориентации.

3. Нажмите *Применить*.

Проверка синхронизации систем

После того как вы загрузили профили пользователей в Sigur, убедитесь, что они были синхронизированы с FindFace Multi. Для этого выполните следующие действия:

1. В FindFace Multi перейдите в список наблюдения $id = 1$. В нем должны отображаться все пользователи Sigur.
2. Отфильтруйте досье по наличию лица. Если синхронизация была выполнена успешно и все данные верные, не должно быть досье без лиц.

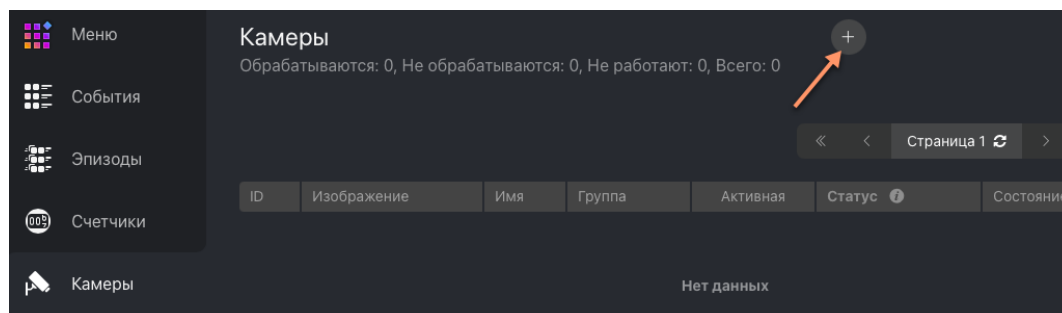
4.4 Периферийные устройства

Существует возможность интеграции FindFace Multi с периферийными устройствами, предоставляющими кадры для распознавания объектов, например, с терминалами контроля доступа. В этом случае при получении кадра от периферийного устройства FindFace Multi будет автоматически инициировать извлечение вектора признаков объекта и создание события. Работа с данными событиями выполняется по аналогии с *событиями* с камер видеонаблюдения.

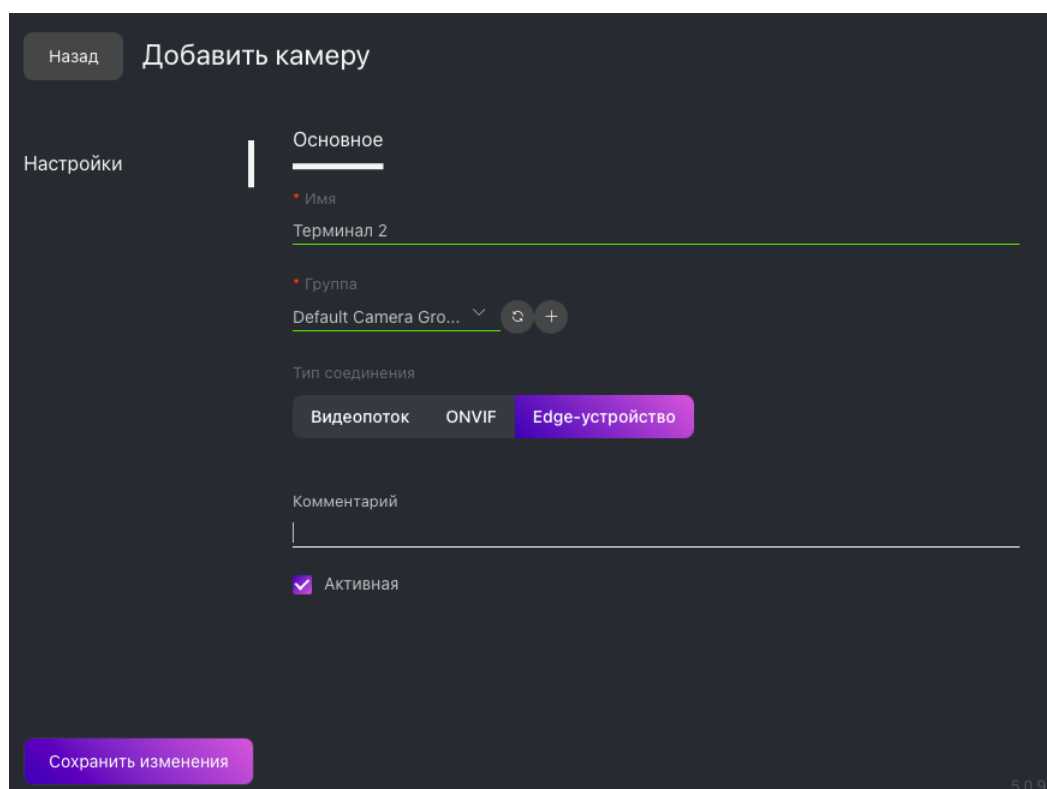
Интеграция периферийных устройств выполняется через *HTTP API*. После первичной настройки FindFace Multi выдаст токен. Указывайте данный токен в каждом API-запросе, отправляемом периферийным устройством в FindFace Multi, чтобы авторизовать устройство.

Для интеграции периферийного устройства с FindFace Multi выполните следующие действия:

1. Перейдите на вкладку *Камеры*.
2. Нажмите $+$.



3. На вкладке *Настройки* -> *Основное* введите общие данные периферийного устройства:

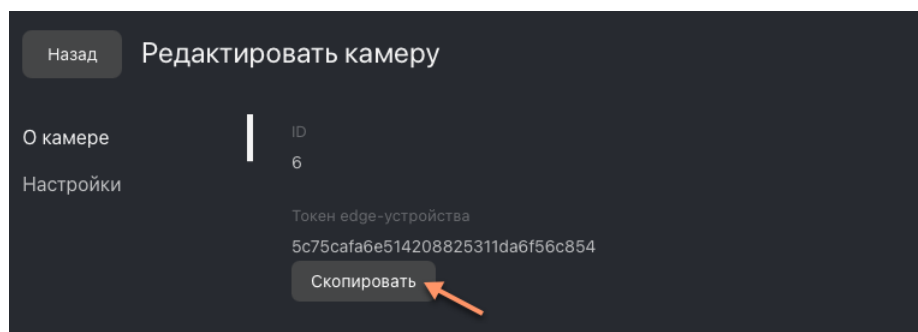


- *Тип соединения*: выберите *Edge-устройство*.
- Укажите имя периферийного устройства.
- Добавьте периферийное устройство в группу камер, чтобы впоследствии было удобнее фильтровать события с данного устройства.

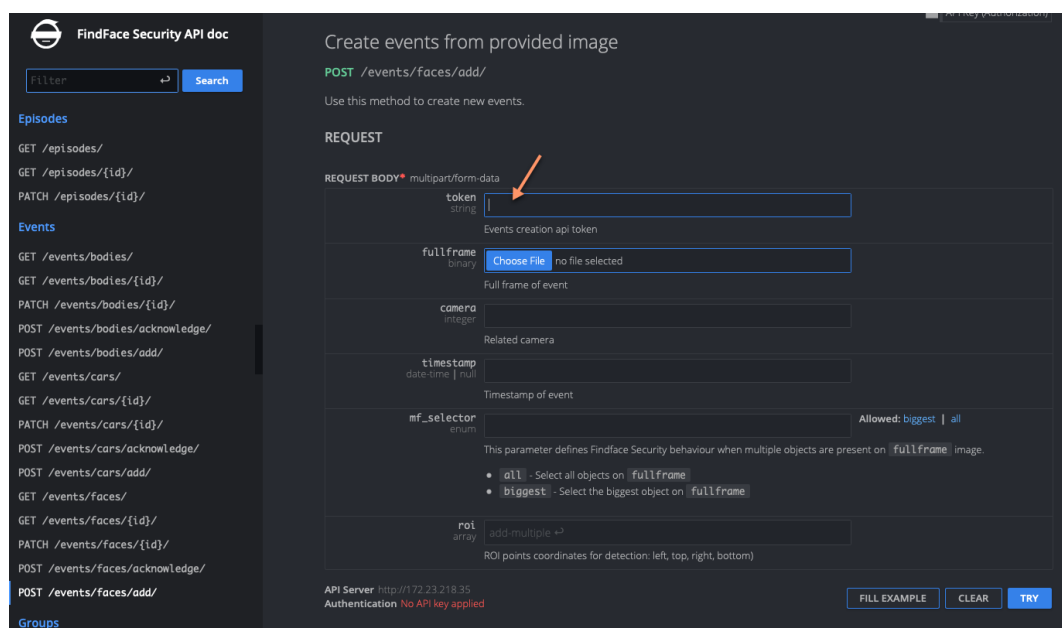
Совет: Вы можете выделить отдельную группу камер специально под периферийные устройства.

- При необходимости добавьте комментарий.
- Поставьте флажок *Активная*.

4. На вкладке *О камере* скопируйте в буфер предоставленный токен периферийного устройства.



5. Указывайте данный токен в каждом API-запросе, отправляемом периферийным устройством в FindFace Multi для создания события. В результате кадры, переданные в запросах, будут связаны с камерой, соответствующей периферийному устройству, и обработаны по аналогии с кадрами с камер видеонаблюдения.



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*.

f

`facrouter.plugin`, [165](#)

n

`ntech.sfapi_client.client`, [168](#)

`ntech.sfapi_client.filters`, [171](#)

`ntech.sfapi_client.gallery`, [169](#)

O

`objects`, [167](#)

СИМВОЛЫ

- `__init__()` (метод `ntech.sfapi_client.filters.Detection`), 173
- `__init__()` (метод `ntech.sfapi_client.filters.Face`), 174
- ### A
- `add()` (метод `ntech.sfapi_client.gallery.Gallery`), 169
- ### B
- `BBox` (класс в `objects`), 167
- ### C
- `Client` (класс в `ntech.sfapi_client.client`), 168
- `create()` (метод `ntech.sfapi_client.gallery.Gallery`), 170
- ### D
- `delete()` (метод `ntech.sfapi_client.gallery.Gallery`), 170
- `detect()` (метод `ntech.sfapi_client.client.Client`), 168
- `Detection` (класс в `ntech.sfapi_client.filters`), 173
- `drop()` (метод `ntech.sfapi_client.gallery.Gallery`), 171
- ### F
- `Face` (класс в `ntech.sfapi_client.filters`), 174
- `facrouter.plugin` модуль, 165
- `Filter` (класс в `ntech.sfapi_client.filters`), 171
- ### G
- `Gallery` (класс в `ntech.sfapi_client.gallery`), 169
- `gallery()` (метод `ntech.sfapi_client.client.Client`), 168
- `get()` (метод `ntech.sfapi_client.gallery.Gallery`), 170
- `gte()` (метод класса `ntech.sfapi_client.filters.Id`), 172
- `gte()` (метод класса `ntech.sfapi_client.filters.Meta`), 173
- ### I
- `Id` (класс в `ntech.sfapi_client.filters`), 172
- ### L
- `list()` (метод `ntech.sfapi_client.gallery.Gallery`), 169
- `lte()` (метод класса `ntech.sfapi_client.filters.Id`), 172
- `lte()` (метод класса `ntech.sfapi_client.filters.Meta`), 172
- ### M
- `Meta` (класс в `ntech.sfapi_client.filters`), 172
- ### N
- `ntech.sfapi_client.client` модуль, 168
- `ntech.sfapi_client.filters` модуль, 171
- `ntech.sfapi_client.gallery` модуль, 169
- ### O
- `objects` модуль, 167
- `objects.DetectFace` (класс в `objects`), 167
- `objects.DetectResponse` (класс в `objects`), 167
- `objects.Face` (класс в `objects`), 167
- `objects.FaceId` (класс в `objects`), 167
- `objects.ListResponse` (класс в `objects`), 167
- `oneof()` (метод класса `ntech.sfapi_client.filters.Id`), 172
- `oneof()` (метод класса `ntech.sfapi_client.filters.Meta`), 173

P

`Plugin` (класс в `facrouter.plugin`), 165
`preprocess()`, 163
`preprocess()` (метод `facrouter.plugin.Plugin`), 165
`process()`, 164
`process()` (метод `facrouter.plugin.Plugin`), 166

S

`serialize()` (метод
 `ntech.sfapi_client.filters.Filter`), 171
`sfapi_client.SFApiMalformedResponseError`
 (класс в `ntech.sfapi_client.filters`), 175
`sfapi_client.SFApiRemoteError` (класс в
 `ntech.sfapi_client.filters`), 174
`shutdown()`, 164
`shutdown()` (метод `facrouter.plugin.Plugin`), 166
`subset()` (метод класса
 `ntech.sfapi_client.filters.Meta`), 173

U

`update()` (метод `ntech.sfapi_client.gallery.Gallery`),
 171

модуль

`facrouter.plugin`, 165
 `ntech.sfapi_client.client`, 168
 `ntech.sfapi_client.filters`, 171
 `ntech.sfapi_client.gallery`, 169
 `objects`, 167