
FindFace Multi

Выпуск 1.0

NtechLab

янв. 18, 2024

1	Руководство системного администратора	5
1.1	Архитектура	5
1.2	Требования	11
1.3	Лицензирование	16
1.4	Развертывание FindFace Multi	18
1.5	Обслуживание и устранение неисправностей	42
1.6	Приложения	70
2	Руководство пользователя	107
2.1	Первые шаги после установки	107
2.2	Работа с FindFace Multi	111
2.3	Расширенный функционал	197
3	Руководство по интеграции	225
3.1	HTTP API	225
3.2	Вебхуки	225
3.3	Интеграции с партнерами	233
	Содержание модулей Python	253
	Алфавитный указатель	255

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 цвета одежды (верх и низ).
- 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.
- Комплексные досье, содержащие сборную информацию о человеке: фотографии его лица, силуэта, автомобиля, автомобильный номер.
- Быстрый и надежный мониторинг видео в реальном времени по базам данных досье.
- 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 и точной настройки камеры под распознавание объектов каждого типа.
- 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 персон.
- AI-распознавание цвета одежды.
- AI-распознавание марки, модели, типа кузова, цвета, номера автомобиля.
- Поиск лиц, силуэтов, автомобилей в базах данных.
- Возможность подсчета лиц и силуэтов на подключенных камерах (подсчет по одной или многим камерам).

- Видеонаблюдение.

Окружающая среда

- 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.
- Распределение базы данных досье между несколькими серверами с синхронизацией и репликацией.
- Возможность лицензирования в открытых и закрытых системах.
- CPU- и GPU-ускорение на ваш выбор.
- Мобильное приложение.

Безопасность

- Расширенное управление пользователями.
- Аутентификация на основании пароля, сертификата и лица для гарантированной защиты системы.
- Безопасность данных в досье.
- Комплексный, удобный журнал действий пользователя с возможностью поиска.
- Утилиты резервного копирования и восстановления.

Расширенное использование системы

- Анализ социальных взаимодействий.
- Аналитика “Знай своего клиента” (KYC).
- Детализированные отчеты на события распознавания лиц, эпизоды, события поиска, персоны, камеры, досье, журнал действий пользователей и аналитику KYC.
- Детектор живых лиц (liveness) как автономный сервис.

Полезные мелочи

- Быстрое создание базы данных досье.
- Полная настройка досье.
- Поддержка дедубликации событий и досье.
- Защита персональных данных (согласно GDPR и аналогичным законам).
- Расширенный набор поисковых фильтров.
- Очистка базы данных по расписанию.

Интеграция

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

Руководство системного администратора

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

1.1 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 развертывания, 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*
- *Развертывание на одиночном сервере или в кластере*
- *Аппаратное ускорение на CPU и GPU*

1.1.1 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*.

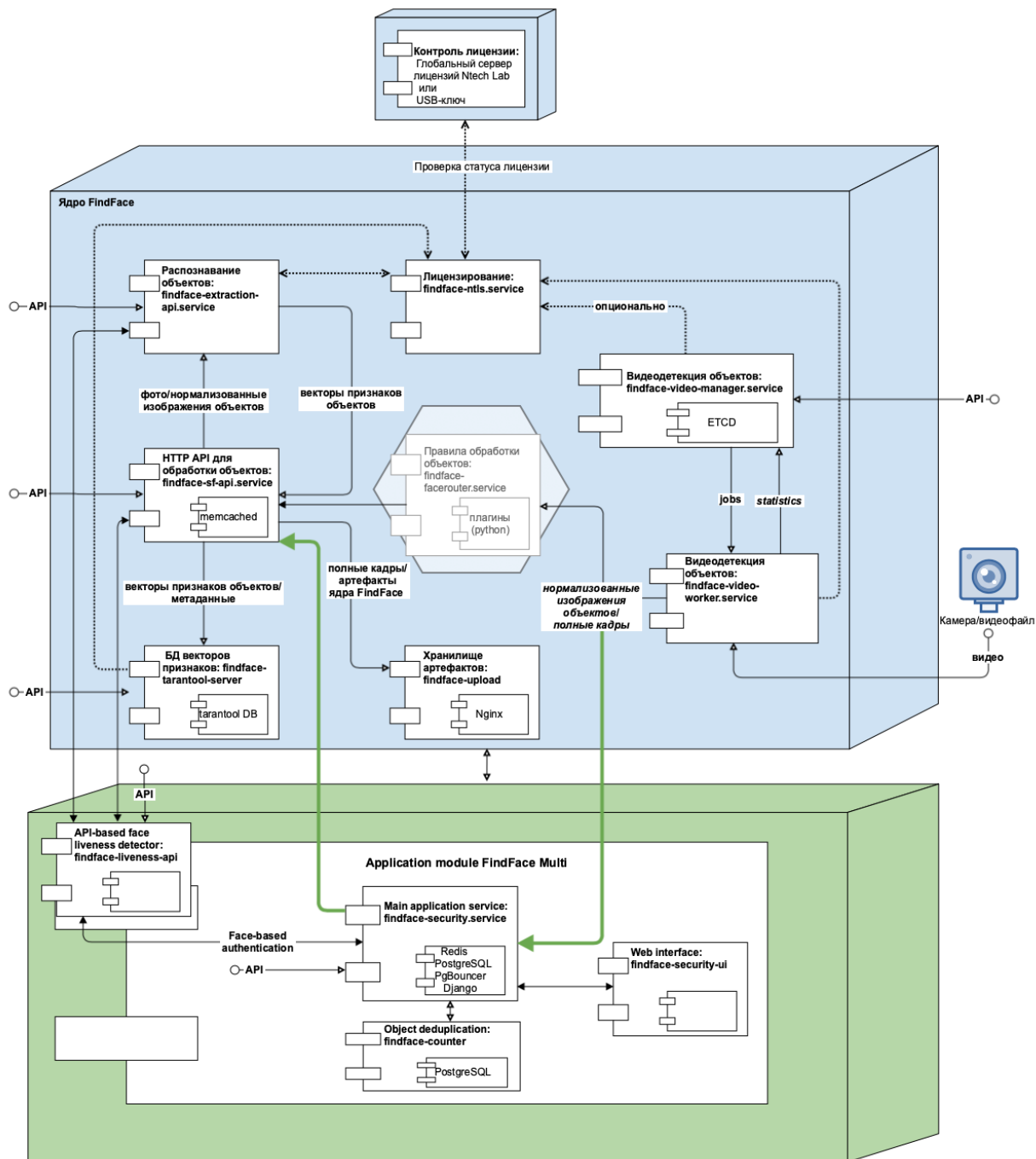
FindFace Multi обнаруживает объект на фотографии или видеоизображении и подготавливает его изображение с помощью нормализации. Далее нормализованное изображение используется для извлечения вектора признаков объекта (n-мерного вектора числовых признаков, которые характеризуют объект). Векторы признаков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

1.1.2 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.

Схема архитектуры



Ядро FindFace

Ядро FindFace включает в себя следующие компоненты:

Ком- по- нент	Исполь- 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	2001, 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.	
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 и 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. Обеспечивает <i>распознавание живых лиц</i> при соответствующих настройках. Поддерживается CPU- и GPU-ускорение.	Tarantool
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 работа с аппаратными лицензионными ключами. Поддерживается лицензирование через прокси-сервер.	
Tarantool	Порты шардов (по умолчанию 330xx, 81xx)	Стороннее программное обеспечение, на основе которого реализована база данных, хранящая извлеченные векторы признаков и события идентификации объектов. Системные данные, досье, пользовательские аккаунты и настройки камер хранятся в PostgreSQL (часть прикладного модуля FindFace Multi).	
etcd	2379	Стороннее программное обеспечение, реализующее распределенное хранилище ключей для компонента findface-video-manager . Используется в качестве координационной службы в распределенной системе, обеспечивая отказоустойчивость видеодетектора объектов.	etcd
NginX	80; SSL: 8002, 8003, 443, 80	Стороннее программное обеспечение, которое реализует веб-интерфейсы системы.	nginx
memcached	11211	Стороннее программное обеспечение, реализующее сервис кэширования данных в оперативной памяти на основе хеш-таблицы. Используется для временного хранения извлеченных векторов признаков перед их записью в базу данных Tarantool.	memcached

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, постановку в очередь видеофайлов на обработку. Включает в себя следующие внутренние сервисы: проверка лицензии, менеджер счетчиков, менеджер вебхуков, группировщик персон, 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
findface- security- ui	На- страи- ва- ет- ся	Основной веб-интерфейс, использующийся для взаимодействия с FindFace Multi. Позволяет работать с событиями идентификации объектов, искать объекты в базах данных, управлять камерами, пользователями, досье и списками наблюдения, собирать статистику в реальном времени и многое другое.	
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 видеофрагмента и возвращает изображение лица наилучшего качества и десятичный результат liveness, усредненный по взятым кадрам. Сервис также используется для аутентификации на базе распознавания лиц. См. <i>Liveness как автономный сервис</i> и <i>Аутентификация и мониторинг сессии пользователя</i> .	
PostgreSQL	5432	Стороннее программное обеспечение, реализующее основную базу данных системы. В данной базе хранятся детализированные досье объектов (лица, силуэты, автомобили) с разбиением по категориям (спискам наблюдения), а также данные внутреннего характера, такие как профили пользователей FindFace Multi, настройки видеокамер и пр. Векторы признаков объектов и события идентификации объектов хранятся в Tarantool (часть ядра FindFace).	PostgreSQL
Pgbouncer	5439	Стороннее программное обеспечение, обеспечивающее облегченный пул соединений для PostgreSQL. Необязательный компонент, используется для увеличения производительности базы данных при высокой нагрузке.	PgBouncer
Redis	6379	Стороннее программное обеспечение, которое реализует брокер сообщений внутри findface-security .	Redis
Django	5439	Стороннее программное обеспечение, реализующее веб-фреймворк для веб-интерфейса FindFace Multi.	Django

См. также:

Подробнее о компонентах

1.1.3 Развертывание на одиночном сервере или в кластере

Вы можете развернуть FindFace Multi как на одиночном сервере, так и в кластерной 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`).
- Полностью распределенная архитектура FindFace Multi. Может потребоваться балансировка нагрузки.

Подробнее см. раздел *Типичная установка в кластере*.

1.1.4 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.

1.2 Требования

В этой главе:

- Системные требования для базовой конфигурации
- Необходимые навыки администратора
- Требования к камерам видеонаблюдения
 - Распознавание лиц
 - Распознавание силуэтов и автомобилей

1.2.1 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)	
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.

1.2.2 Необходимые навыки администратора

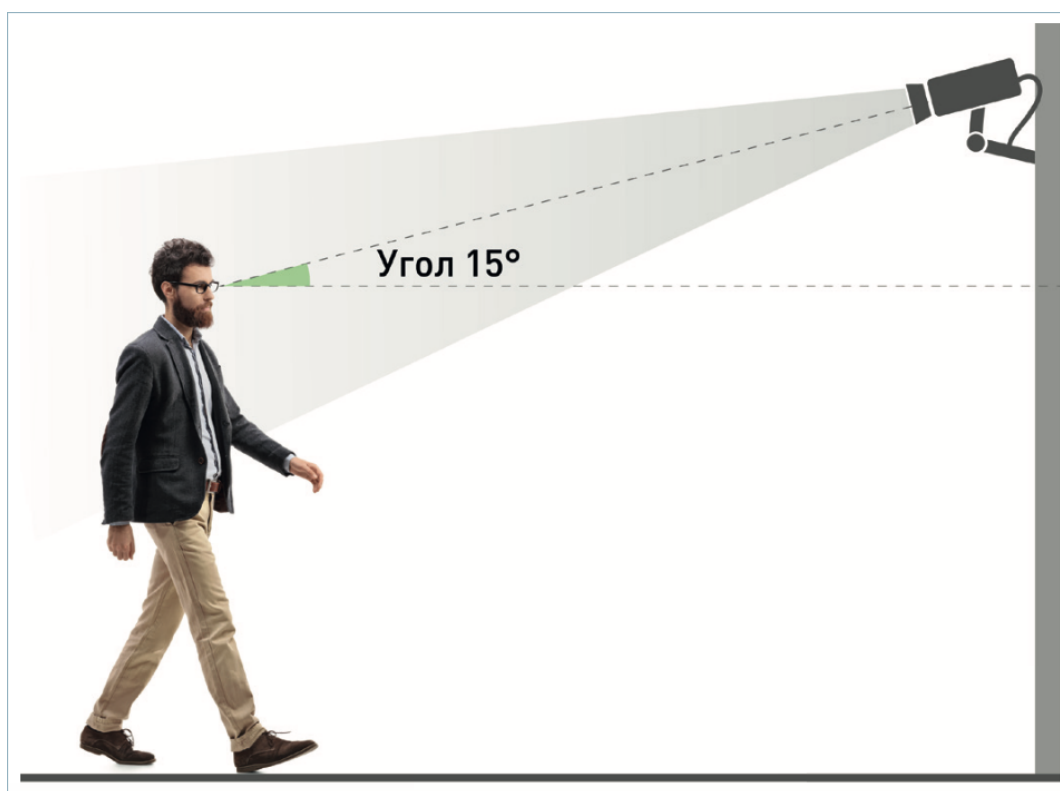
Администратор FindFace Multi должен обладать навыками работы с ОС Ubuntu на уровне продвинутого пользователя.

1.2.3 Требования к камерам видеонаблюдения

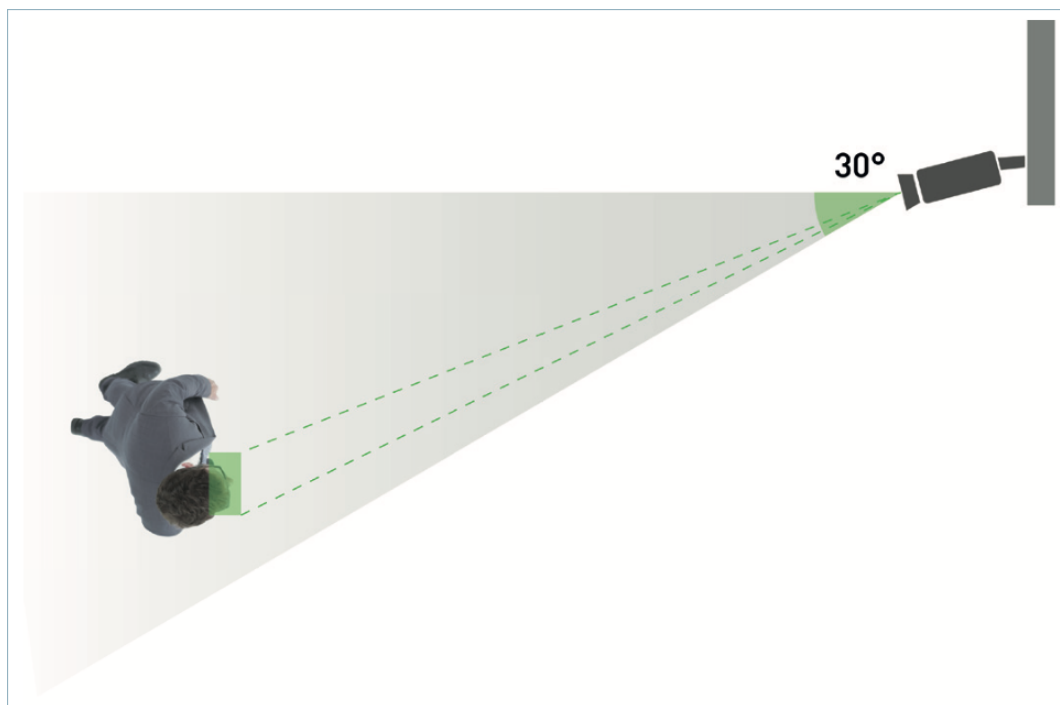
Распознавание лиц

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).



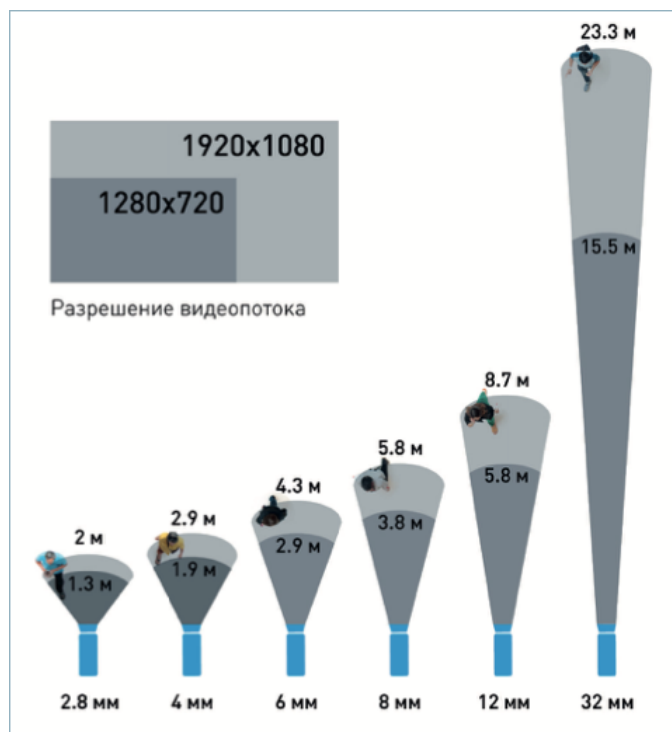
3. Угол горизонтального отклонения не должен превышать 30° . Горизонтальное отклонение — это отклонение оптической оси видеокамеры от вектора движения основного потока объектов распознавания.



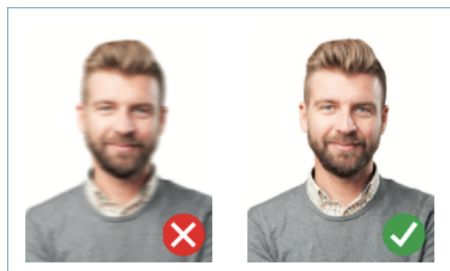
4. Минимальная плотность пикселей для идентификации – 500 пикселей/м (примерно соответствует ширине лица 80 пикселей).



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 — это новые функции, впервые представленные в текущей версии, 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).

1.3 Лицензирование

В этой главе:

- Принцип лицензирования
- Просмотр и обновление лицензии

1.3.1 Принцип лицензирования

FindFace Multi лицензируется по следующим критериям:

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

Примечание: Количество векторов признаков, извлеченных из объектов, обнаруженных на видео, из загружаемых в досье фотографий и при построении т. н. центроидов в процессе *классификации персон*.

Схема лицензирования выглядит следующим образом:

- События: 1 событие распознавания объекта на видео = 1 объект в лицензии.
 - Досье: 1 фотография в досье = 1 объект в лицензии.
 - Персоны: 1 персона = 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 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 онлайн-лицензированием и лицензированием в закрытой сети:

- 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.

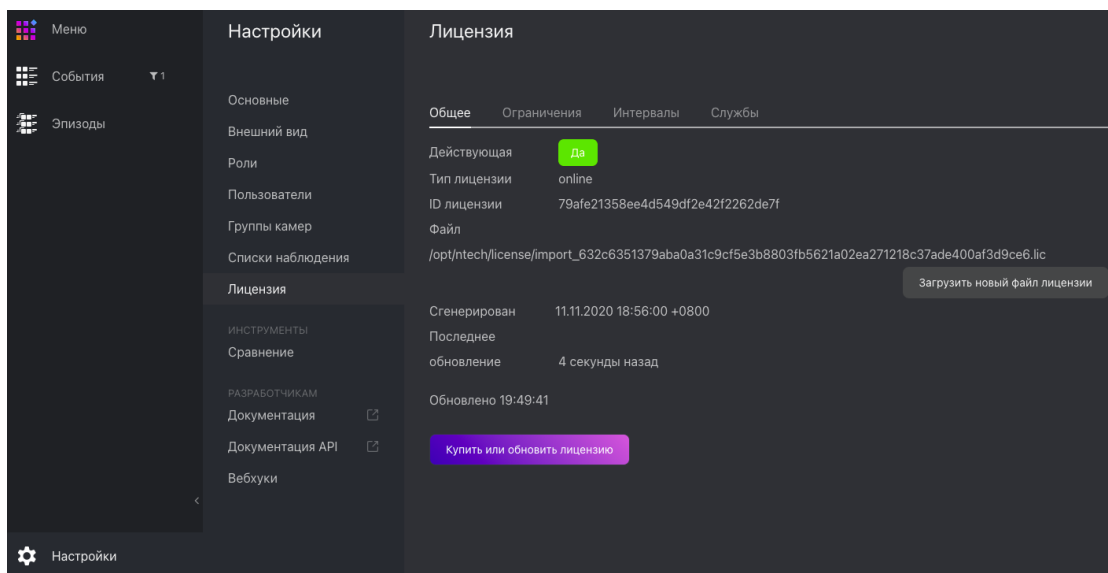
Примечание: Уведомите своего менеджера, если требуется продлить данный период (до 2-х дней).

- Для лицензирования в закрытой сети необходимо наличие USB-порта на физическом сервере с компонентом `findface-ntls` (сервер лицензирования в составе *ядра FindFace*) для подключения предоставляемого USB-ключа аппаратной защиты.

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.

1.3.2 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.

См.также:

Устранение неполадок с лицензированием и findface-ntls

1.4 Развертывание FindFace Multi

FindFace Multi предоставляет следующие варианты развертывания:

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

Важно: Первый после развертывания запуск сервисов `findface-extract-api-gpu` и `findface-video-worker-gpu` может занять продолжительное время, до 45 минут, из-за процесса кэширования.

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.

1.4.1 Развертывание из консольного инсталлятора

Для развертывания FindFace Multi используется консольный инсталлятор.

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

Важно: 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-1.0-and-server-5.0.1.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-1.0-and-server-5.0.1.run
```

4. Запустите файл `.run`.

```
sudo ./findface-multi-1.0-and-server-5.0.1.run
```

Инсталлятор задаст вам несколько вопросов, после чего проверит, соответствует ли сервер системным требованиям. Вопросы 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 только apt-репозиторий для *пошагового развертывания* в будущем.

Важно: При данном типе установки модели нейронных сетей, необходимые для функционирования `findface-extraction-api`, не устанавливаются. Обязательно установите их *вручную* на серверах с `findface-extraction-api`.

- 4: полностью настраиваемая установка (установка нужных пакетов).

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

3. Тип пакета `findface-video-worker`: CPU или GPU.

4. Тип пакета `findface-extraction-api`: CPU или GPU.

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

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 FindFace Security до FindFace Multi*.

Сер-вис	Конфигурация
postgresql-10	Устанавливается и запускается.
redis-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.
findface-sf-api	Устанавливается и запускается.
findface-upload	Устанавливается.
findface-video-manager	Устанавливается и запускается (CPU/GPU-ускорение).
findface-video-worker-*	Устанавливается и запускается.
findface-data-*	Модели нейронных сетей для распознавания объектов и их атрибутов. Устанавливаются.
findface-gpudetector-data	Данные gpu-детектора NTechLab. Устанавливается.
python3-ntech.ffsecurity-client	Библиотека клиента Python, используемая в API FindFace Multi. Устанавливается.
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.

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/
```

- 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: Для справки см. [данную таблицу](#).

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

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

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

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

```
sudo ./findface-multi-1.0-and-server-5.0.1.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*.

1.4.2 Пошаговое развертывание из apt-репозитория

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

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-1.0-and-server-5.0.1.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-1.0-and-server-5.0.1.run
```

4. Запустите файл `.run`.

```
sudo ./findface-multi-1.0-and-server-5.0.1.run
```

Инсталлятор задаст вам несколько вопросов, после чего проверит, соответствует ли сервер системным требованиям. Вопросы 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`. Введите `done` для сохранения выбора и перехода к следующему шагу.

Важно: Должна быть установлена как минимум одна модель для биометрии лица.

После этого apt-репозиторий FindFace будет автоматически установлен.

Установка необходимого стороннего ПО

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

1. Установите пакеты с указанным сторонним ПО следующим образом:

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

2. Откройте файл конфигурации `/etc/memcached.conf`. Установите максимальный размер памяти в мегабайтах, используемый для хранения элементов: `-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
```

(continues on next page)

(продолжение с предыдущей страницы)

```
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 запустите их:

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

Обеспечение лицензирования

См. также:

Лицензирование

Вы получаете файл лицензии вместе с установочными пакетами FindFace Multi. Для лицензирования в закрытой сети вам также будет предоставлен ключ аппаратной защиты.

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*.
2. Установите компоненты ядра FindFace:

```

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

```

Примечание: Для того чтобы установить компонент `findface-extraction-api` с GPU-ускорением, вместо `findface-extraction-api` в команде введите `findface-extraction-api-gpu`.

Примечание: Для того чтобы установить компонент `findface-video-worker` с GPU-ускорением, вместо `findface-video-worker-cpu` в команде введите `findface-video-worker-gpu`. 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
...
```

- Откройте файл конфигурации `/etc/findface-video-worker-cpu.ini` (`/etc/findface-video-worker-gpu.ini`). 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.

```
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
```

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

```
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
```

Развертывание прикладного модуля FindFace Multi и базы данных векторов признаков

Для развертывания прикладного модуля FindFace Multi, выполните следующие действия:

- 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
```

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

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 использует для работы с данным функционалом **EXTERNAL_ADDRESS**. Для использования Genetec и вебхуков обязательно укажите по крайней мере один из параметров **SERVICE_EXTERNAL_ADDRESS** / **EXTERNAL_ADDRESS**.
- (Опционально) 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 **EXTERNAL_ADDRESS**.
- **VIDEO_DETECTOR_TOKEN:** придумайте токен и укажите его в данном параметре, чтобы авторизовать модуль видеодетекции лиц.
- **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 лица 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 перенадресации* 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 обнаруженное лицо отсутствует в списках наблюдения (верификация дала отрицательный результат). Данную настройку рекомендуется использовать при большом количестве посетителей. Пороговая степень схожести при верификации лиц определяется параметром **CONFIDENCE_THRESHOLD**.

Совет: Не рекомендуется редактировать значение параметра **MINIMUM_DOSSIER_QUALITY**. Данный параметр определяет минимальное качество лица на фотографии в досье. Если качество лица хуже минимального, пользователь не сможет загрузить такую фотографию в досье. Прямые изображения лиц анфас считаются наиболее качественными.

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. Обеспечьте лицензирование. Используйте веб-интерфейс FindFace Multi, чтобы *загрузить лицензионный файл*, который вы до этого получили от своего менеджера (*Настройки -> Лицензия*). Для лицензирования в закрытой системе вставьте ключ аппаратной защиты в USB-порт.

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

1.4.3 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
...
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-security-and-server-5.0.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-security-and-server-5.0.run
```

4. Запустите файл `.run`.

```
sudo ./findface-security-and-server-5.0.run
```

Инсталлятор задаст вам несколько вопросов, после чего проверит, соответствует ли сервер системным требованиям. Вопросы 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: Ответы на вопросы будут сохранены в файл `/tmp/<findface-installer-*>.json`. Отредактируйте его и используйте для установки `findface-video-worker` на других серверах, не отвечая повторно на вопросы инсталлятора.

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-security-and-server-5.0.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*.

1.4.4 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-1.0-and-server-5.0.1.run`.

```
sudo ./findface-multi-1.0-and-server-5.0.1.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*.

1.4.5 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`.

1.4.6 Типичная установка в кластере

Данный раздел посвящен разворачиванию FindFace Multi в кластерной среде.

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

Разворачивание FindFace Multi в кластере 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 разворачивания в кластере. В высоконагруженных системах также может потребоваться распределить обработку API-запросов, т. е. организовать несколько серверов `findface-sf-api` и `findface-video-manager`. В этом случае руководствуйтесь инструкциями в разделе *Полностью настраиваемая установка*.

Разворачивание FindFace Multi в кластерной среде состоит из следующих этапов:

- *Разворачивание центрального сервера*
- *Разворачивание серверов обработки видео*
- *Разворачивание серверов экстракции*
- *Распределение нагрузки между серверами экстракции*

- Организация распределенной 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 через веб-интерфейс центрального сервера **http://<IP_адрес_сервера>/#/license**. Для доступа в веб-интерфейс используйте логин и пароль администратора, 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-1.0-and-server-5.0.1.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: `1 7`. 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. Для того чтобы выбрать модели, сначала снимите выделение со всех компонентов, введя в командной строке `-*`, затем введите порядковые номера компонентов, например, `8 2 7`. Введите `done` для сохранения выбора и перехода к следующему шагу. 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-1.0-and-server-5.0.1.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;
    }
}
```

(continues on next page)

(продолжение с предыдущей страницы)

```
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.

Организация распределенной базы данных

Компонент `findface-tarantool-server` соединяет базу данных векторов признаков Tarantool и компонент `findface-sf-api`, передавая результаты поиска от базы данных в `findface-sf-api` для дальнейшей обработки. 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. Каждый шард может обрабатывать приблизительно до 10 000 000 объектов. При разворачивании `findface-tarantool-server` из инсталлятора шарды создаются автоматически с учетом аппаратной конфигурации физического сервера.

Для того чтобы распределить базу данных векторов признаков, установите `findface-tarantool-server` на каждом сервере базы данных. 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. Ответы на вопросы инсталлятора будут сохранены в файл `/tmp/<findface-installer-*>.json`. Используйте данный файл, чтобы установить `findface-tarantool-server` на других серверах, не отвечая на вопросы инсталлятора повторно.

```
sudo ./findface-multi-1.0-and-server-5.0.1.run -f /tmp/<findface-installer-*>.json
```

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.

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

```
...
extraction-api:
  extraction-api: http://172.168.1.9:18667
```

(continues on next page)

(продолжение с предыдущей страницы)

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

```
sudo vi /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'
```

(continues on next page)

(продолжение с предыдущей страницы)

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

На этом установка FindFace Multi в кластерной 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*.

1.4.7 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:

1. Загрузите файл инсталлятора `findface-multi-1.0-and-server-5.0.1.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-1.0-and-server-5.0.1.run
```

4. Запустите файл `.run`.

```
sudo ./findface-multi-1.0-and-server-5.0.1.run
```

5. Выберите продукт для установки: NVIDIA CUDA drivers.

1.5 Обслуживание и устранение неисправностей

1.5.1 Обновление FindFace Security до FindFace Multi

Если вы уже используете наш продукт FindFace Security, вы можете обновить его до FindFace Multi.

Важно: Текущая лицензия будет определять объем функций, активированных в FindFace Multi после обновления.

Предупреждение: Имейте в виду, что при обновлении с FindFace Security 4.2 или более ранней версии *видеоархивы* будут утеряны. Вы не сможете их повторно обработать с другими параметрами в FindFace Multi. Если вы используете данную функциональность, сначала удалите видео в веб-интерфейсе FindFace Security. После обновления повторно загрузите и обработайте те видео, которые вы хотите сохранить, в FindFace Multi.

Если вы этого не сделаете, то после обновления вы по-прежнему сможете видеть события по видеоархивам, но уже без привязки к определенному архиву.

Вы можете найти видеоархивы в директории `/var/lib/findface-security/uploads/videos/` (путь по умолчанию). Действительный путь к директории указан в параметре `MEDIA_ROOT` в файле `/etc/findface-security/conf.py`. Рекомендуется удалить данные видео, если они вам больше не нужны.

Для обновления FindFace Security с любой предыдущей версии до FindFace Multi 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`.

```
# FindFace Security 4.2 and earlier

sudo vi /etc/ffsecurity/config.py

# FindFace Security 4.3 and later

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',
```

(continues on next page)

(продолжение с предыдущей страницы)

```
...
'ROUTER_URL': 'http://172.20.77.58',
...
}
```

2. Остановите сервис `findface-security`.

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

3. 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 хранилищ данных*.

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

4. Установите apt-репозиторий с версией FindFace Multi, используя консольный инсталлятор согласно инструкции в *этом разделе*.
5. Для обновления с FindFace Security 4.1.2 или более ранней версии выполните следующие действия:

1. 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
```

2. Установите пакет `pgbouncer` следующим образом:

```
sudo apt update
sudo apt install -y pgbouncer
```

3. 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
```

(continues on next page)

(продолжение с предыдущей страницы)

```
max_client_conn = 16384
default_pool_size = 20
syslog = 1
```

4. Скопируйте пароль пользователя `ntech` (`9T3g1nXy9yx3y8MIGm9fbef3dia8UTc3` в примере). В PostgreSQL установите скопированный пароль для роли `ntech`. Откройте интерактивный терминал PostgreSQL. Появится строка `postgres=#`. После знака `#` введите следующую команду: `ALTER ROLE ntech PASSWORD '<скопированный пароль>'`.

```
sudo -u postgres psql

postgres=# ALTER ROLE ntech PASSWORD '9T3g1nXy9yx3y8MIGm9fbef3dia8UTc3';
```

5. Через интерактивный терминал PostgreSQL создайте базу данных `ffcounter` в PostgreSQL.

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

6. Перезапустите PostgreSQL.

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

7. Создайте и настройте `pgbouncer.service`.

```
sudo touch /etc/systemd/system/pgbouncer.service
sudo vi /etc/systemd/system/pgbouncer.service
```

Вставьте следующий код:

```
[Unit]
Description=Pgbouncer service
After=postgresql.service
Before=findface-security.service
[Service]
User=postgres
Group=postgres
ExecStart=/usr/sbin/pgbouncer "/etc/pgbouncer/pgbouncer.ini"
[Install]
WantedBy=multi-user.target
```

8. Добавьте `pgbouncer.service` в автозагрузку и перезапустите сервис:

```
sudo systemctl enable pgbouncer.service
sudo systemctl restart pgbouncer.service
```

6. 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`.

7. Добавьте сервисы `findface-counter` и `findface-liveness-api` в автозагрузку.

```
sudo systemctl enable findface-counter.service findface-liveness-api.service
```

8. 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',
```

9. 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`. Убедитесь, что все пользовательские параметры из старой версии присутствуют в новой. 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
sudo vi /etc/findface-extraction-api.ini.ucf-old
sudo vi /etc/findface-extraction-api.ini

...
```

10. Перезагрузите сервисы.

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

11. 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
```

12. 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}/*
```

13. 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
})
```

14. 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
```

15. Восстановите базу данных Tarantool из резервной копии.

```
cd /etc/findface_dump
```

(continues on next page)

(продолжение с предыдущей страницы)

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

16. Выполните миграцию старых данных с версии API v2 на v3 сервиса `findface-sf-api`.

```
sudo findface-sf-api-migrate-v2-v3 -config /etc/findface-sf-api.ini
```

17. 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
```

18. Перезапустите сервис `findface-security`.

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

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
```

1.5.2 Резервное копирование и восстановление хранилищ данных

Этот раздел посвящен резервному копированию и восстановлению хранилищ данных FindFace Multi. Система использует следующие хранилища:

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

В этом разделе:

- Резервное копирование и восстановление базы данных векторов признаков
 - Утилиты
 - Резервное копирование базы данных
 - Восстановление базы данных
- Резервное копирование и восстановление основной базы данных
- Резервное копирование и восстановление артефактов

Резервное копирование и восстановление 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 силуэтов в фотографиях досье.
- `ffsec_car_events`: векторы признаков, извлеченные из обнаруженных на видео автомобилей.
- `ffsec_car_objects`: векторы признаков, извлеченные из автомобилей в фотографиях досье.
- `ffsec_face_events`: векторы признаков, извлеченные из обнаруженных на видео лиц.
- `ffsec_face_objects`: векторы признаков, извлеченные из лиц в фотографиях досье.
- `ffsec_user_face`: векторы признаков, извлеченные из фотографий пользователей FindFace Multi, используемые для аутентификации по лицу.
- `ffsec_persons`: центроиды персон (виртуальные векторы признаков, усредненные по всем лицам персоны) и метаданные.

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 досье 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

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

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/`

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
```

1.5.3 Миграция на другую модель биометрического образца

Совет: Не стесняйтесь обращаться к нашим специалистам по вопросам миграции по адресу support@ntechlab.com.

В процессе эксплуатации системы (например, при обновлении до последней версии продукта) вам может потребоваться миграция биометрических данных на другую модель нейронной сети.

Для миграции 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 хранилищ данных*.

```
mkdir -p /etc/findface_dump
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.ini` и замените прежнюю модель нейронной сети на новую (`kiwi_320.cpu.fnk` в примере).

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

face: face/jackfruit_480.cpu.fnk -> face: face/kiwi_320.cpu.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
  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:
```

(continues on next page)

(продолжение с предыдущей страницы)

```
endpoint: ""
bucket-name: ""
access-key: ""
secret-access-key: ""
secure: False
region: ""
public-url: ""
operation-timeout: 30
```

В разделе `storage-api-from` укажите шарды, с которых осуществляется миграция.

```
storage-api-from: # current location of the gallery
...
shards:
  - master: http://127.0.0.1:8101/v2/
    slave: ""
  - master: http://127.0.0.1:8102/v2/
    slave: ""
  - master: http://127.0.0.1:8103/v2/
    slave: ""
  - master: http://127.0.0.1:8104/v2/
...

```

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 досье, 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: ""

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 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
```

12. Перезагрузите сервисы.

```
sudo systemctl restart findface-security.service
sudo systemctl restart findface-extraction-api findface-video-worker* findface-
↪video-manager findface-sf-api
```

1.5.4 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 хранилищ 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. Очистите директории с данными для всех активных шардов.

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

7. 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 /tmp/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
```

8. Запустите сервис findface-security.

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

См.также:

Пользовательские метаданные лиц досье в Tarantool

1.5.5 Удаление экземпляра продукта

Вы можете автоматически удалить FindFace Multi вместе с базой данных с помощью скрипта `ffsec_uninstall.sh`. Перед удалением будут созданы резервные копии файлов конфигурации и базы данных.

Выполните следующие действия:

1. Загрузите скрипт `ffsec_uninstall.sh` в любой каталог на сервере установки (например, в `/home/username/`).
2. Из данного каталога сделайте скрипт исполняемым.

```
chmod +x ffsec_uninstall.sh
```

3. Запустите скрипт.

```
sudo ./ffsec_uninstall.sh
```

4. Ответьте **all** на вопрос интерактивного мастера удаления, чтобы полностью удалить FindFace Multi вместе с базой данных.

1.5.6 Проверка статуса компонентов

Проверьте статус компонентов, если вы столкнулись с проблемой в системе.

Компонент	Команда для просмотра статуса сервиса
findface-extraction-api	sudo systemctl status findface-extraction-api.service
findface-sf-api	sudo systemctl status findface-sf-api.service
findface-tarantool-server	sudo systemctl status tarantool.service
Шарды findface-tarantool-server	sudo systemctl status tarantool@shard-00*
findface-video-manager	sudo systemctl status findface-video-manager.service
findface-video-worker	sudo systemctl status findface-video-worker*.service
findface-ntls	sudo systemctl status findface-ntls
findface-security	sudo systemctl status findface-security.service
findface-counter	sudo systemctl status findface-counter.service
findface-liveness-api	sudo systemctl status findface-liveness-api.service
etcd	sudo systemctl status etcd.service
NginX	sudo systemctl status nginx.service
memcached	sudo systemctl status memcached.service
postgresql	sudo systemctl status postgresql*
redis	sudo systemctl status redis.service
pgbouncer	sudo systemctl status pgbouncer.service

1.5.7 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 К, М, Г, Т, Р, Е 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

1.5.8 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": 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_
b68d7b7ec9a7310d18832035318c990c9ddf11e3a9ab0ae962f9be48645e196d1.lic",
  "limits": [
    {
      "type": "time",
      "name": "end",
```

(continues on next page)

(продолжение с предыдущей страницы)

```
"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",
  "value": true
},
{
  "type": "boolean",
  "name": "fast-index",
  "value": true
},
{
  "type": "boolean",
  "name": "sec-genetec",
  "value": false
},
{
  "type": "boolean",
  "name": "beard",
  "value": false
},
{
  "type": "boolean",
```

(continues on next page)

(продолжение с предыдущей страницы)

```
    "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"
  }
]
}
```

1.5.9 Автоматическое восстановление Tarantool

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-*
```

1.5.10 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 (лица, силуэты, автомобили), для которых были найдены совпадения в базе данных, и связанные эпизоды (в текущей реализации только лица),
- события без совпадений (лица, силуэты, автомобили) и связанные эпизоды (только лица),
- полные кадры событий с совпадениями (лица, силуэты, автомобили),
- полные кадры событий без совпадений (лица, силуэты, автомобили),
- записи счетчика,
- события создания персоны,
- журнал действий пользователей.

Справка по утилите `cleanup` вызывается следующей командой:

```
sudo findface-security cleanup --help
usage: findface-security cleanup [-h] [--as-configured]
                                [--car-events-matched-age CAR_EVENTS_MATCHED_AGE]
                                [--car-events-unmatched-age CAR_EVENTS_UNMATCHED_AGE]
                                [--car-events-fullframe-matched-age CAR_EVENTS_
↪FULLFRAME_MATCHED_AGE]
                                [--car-events-fullframe-unmatched-age CAR_EVENTS_
↪FULLFRAME_UNMATCHED_AGE]
                                [--face-events-matched-age FACE_EVENTS_MATCHED_AGE]
                                [--face-events-unmatched-age FACE_EVENTS_UNMATCHED_AGE]
                                [--face-events-fullframe-matched-age FACE_EVENTS_
↪FULLFRAME_MATCHED_AGE]
                                [--face-events-fullframe-unmatched-age FACE_EVENTS_
↪FULLFRAME_UNMATCHED_AGE]
                                [--body-events-matched-age BODY_EVENTS_MATCHED_AGE]
                                [--body-events-unmatched-age BODY_EVENTS_UNMATCHED_AGE]
                                [--body-events-fullframe-matched-age BODY_EVENTS_
↪FULLFRAME_MATCHED_AGE]
                                [--body-events-fullframe-unmatched-age BODY_EVENTS_
↪FULLFRAME_UNMATCHED_AGE]
                                [--counter-records-age COUNTER_RECORDS_AGE]
                                [--person-events-age PERSON_EVENTS_AGE]
                                [--audit-logs-age AUDIT_LOGS_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
```

(continues on next page)

(продолжение с предыдущей страницы)

```

    and persons. Can't be used with other arguments.
--car-events-matched-age CAR_EVENTS_MATCHED_AGE
    Minimum age in days of matched car events to clean up
--car-events-unmatched-age CAR_EVENTS_UNMATCHED_AGE
    Minimum age in days of unmatched car events to clean
    up
--car-events-fullframe-matched-age CAR_EVENTS_FULLFRAME_MATCHED_AGE
    Minimum age in days of matched car events fullframes
    to clean up
--car-events-fullframe-unmatched-age CAR_EVENTS_FULLFRAME_UNMATCHED_AGE
    Minimum age in days of unmatched car events fullframes
    to clean up
--face-events-matched-age FACE_EVENTS_MATCHED_AGE
    Minimum age in days of matched face events to clean up
--face-events-unmatched-age FACE_EVENTS_UNMATCHED_AGE
    Minimum age in days of unmatched face events to clean
    up
--face-events-fullframe-matched-age FACE_EVENTS_FULLFRAME_MATCHED_AGE
    Minimum age in days of matched face events fullframes
    to clean up
--face-events-fullframe-unmatched-age FACE_EVENTS_FULLFRAME_UNMATCHED_AGE
    Minimum age in days of unmatched face events
    fullframes to clean up
--body-events-matched-age BODY_EVENTS_MATCHED_AGE
    Minimum age in days of matched body events to clean up
--body-events-unmatched-age BODY_EVENTS_UNMATCHED_AGE
    Minimum age in days of unmatched body events to clean
    up
--body-events-fullframe-matched-age BODY_EVENTS_FULLFRAME_MATCHED_AGE
    Minimum age in days of matched body events fullframes
    to clean up
--body-events-fullframe-unmatched-age BODY_EVENTS_FULLFRAME_UNMATCHED_AGE
    Minimum age in days of unmatched body events
    fullframes to clean up
--counter-records-age COUNTER_RECORDS_AGE
    Minimum age in days of counter records to clean up
--person-events-age PERSON_EVENTS_AGE
    Minimum age in days of person events to clean up
--audit-logs-age AUDIT_LOGS_AGE
    Minimum age in days of audit logs to clean up
--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

```

(continues on next page)

(продолжение с предыдущей страницы)

```

used.
--pythonpath PYTHONPATH      A directory to add to the Python path, e.g.
                              "/home/djangoprojects/myproject".
--traceback                  Raise on CommandError exceptions
--no-color                    Don't colorize the command output.
--force-color                 Force colorization of the command output.
--skip-checks                 Skip system checks.

```

Для полного удаления событий и эпизодов старше определенного количества дней используйте аргументы `--events-matched-age`/`--events-unmatched-age`. Например, для удаления событий с автомобилями без совпадений старше 5 дней выполните команду:

```
sudo findface-security cleanup --car-events-unmatched-age 5
```

Для удаления событий распознавания автомобилей с совпадениями старше 5 дней выполните команду:

```
sudo findface-security cleanup --car-events-matched-age 5
```

Следующие команды удаляют только полные кадры событий распознавания силуэтов с совпадениями и без:

```

sudo findface-security cleanup --body-events-fullframe-matched-age 5
sudo findface-security cleanup --body-events-fullframe-unmatched-age 5

```

Для того чтобы удалить только записи счетчика, выполните команду:

```
sudo findface-security cleanup --counter-records-age 5
```

Для того чтобы удалить только события создания персон, выполните команду:

```
sudo findface-security cleanup --person-events-age 5
```

Для того чтобы удалить только записи в журнале действий пользователей, выполните команду:

```
sudo findface-security cleanup --audit-logs-age 5
```

Важно: Должен быть задан хотя бы один аргумент из указанных.

1.5.11 Отключение сервисов

Существует возможность отключить следующие сервисы FindFace Multi, если вы ими не пользуетесь:

- эпизоды
- менеджер очередей обработки видеоархивов
- вебхуки
- персоны

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,
        "persons": False,
    }
}
```

```
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 новые артефакты перестанут поступать.

1.5.12 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
Поиск	"search"
События	"events"
Эпизоды	"episodes"
Персоны	"persons"
Связи	"contacts"
Счетчики	"counters"
Камеры	"cameras"
Видеостена	"video-wall"
Отчеты	"reports"
Журнал действий	"audit_logs"
Досье	"dossiers"
Видео	"videos"
Аналитика	"know_your_customer"
Настройки	"preferences"
Основные настройки	"settings"
Внешний вид	"appearance"
Роли	"roles"
Пользователи	"users"
Группы камер	"camera-groups"
Списки наблюдения	"dossier-lists"
Лицензия	"license"
Инструменты	"tools"
Сравнение	"verify"
Разработчикам	"developer"
Документация	"documentation"
API-документация	"api_doc"
Вебхуки	"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
```

1.5.13 Сброс пароля

Для того чтобы сбросить пользовательский пароль в веб-интерфейсе FindFace Multi, выполните следующую команду:

```
findface-security changepassword %username
```

1.5.14 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. Узнайте текущую директорию базы данных.

```
postgres=# SHOW data_directory;

data_directory
-----
/var/lib/postgresql/10/main
```

2. Остановите PostgreSQL.

```
sudo systemctl stop postgresql
```

3. Создайте новую директорию для базы данных и назначьте ее пользователю `ntech`.

```
mkdir postgres_data_dir
chown ntech postgres_data_dir
```

4. Перенесите базу данных и создайте резервную копию прежней версии.

```
sudo rsync -av /var/lib/postgresql /test/postgres_data_dir/
sudo mv /var/lib/postgresql/10/main /backups/pg_backup
```

5. Замените директорию в файле конфигурации PostgreSQL <директория PostgreSQL>/postgresql.conf.

```
data_directory = '/test/postgres_data_dir/postgresql/10/main'
```

6. Запустите PostgreSQL.

```
sudo systemctl start postgresql
```

7. Убедитесь, что директория успешна изменена.

```
postgres=# SHOW data_directory;

      data_directory
-----
/test/postgres_data_dir/postgresql/10/main
```

1.6 Приложения

1.6.1 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_schema</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> .

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.
motion_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_transpotr=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	Прислать 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),

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. Объекты меньшего размера не отправляются. Значение по умолчанию: 1.
<code>filter_max_size</code>	Определяет максимальный размер объекта в пикселях. Объекты большего размера не отправляются. Значение по умолчанию: 8192.
<code>roi</code>	Включает отправку на Сервер объектов, обнаруженных только внутри интересующей области $W \times H + X + Y$. По умолчанию область не задана.
<code>fullframe_crop_rot</code>	Отправлять отправляемые полные кадры по ROT (региону отслеживания). Значение по умолчанию: false.
<code>fullframe_use_png</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>	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_best_of_interval</code>	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_interpolate_boxes</code>	Восстанавливать 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_interval</code>	Время, в течение которого завершит трек, если в нем не было задетектировано новых изображений объекта в течение заданного времени (секунды). Значение по умолчанию: 1.
<code>track_max_duration_frames</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_fullframe</code>	Отправлять полные кадры для обнаруженных объектов. Значение по умолчанию: true.
<code>post_best_track_normalize</code>	Отправлять нормализованные изображения для обнаруженных объектов. Значение по умолчанию: true.
<code>post_first_track_fullframe</code>	Отправлять первый кадр трека. Значение по умолчанию: false.
<code>post_last_track_fullframe</code>	Отправлять последний кадр трека. Значение по умолчанию: false.

1. 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*

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.
- **stream_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.
- **router_url**: IP-адрес и порт компонента `findface-facerouter` (`findface-security` в стандартной конфигурации FindFace Multi), в который компонент `findface-video-worker` отправляет обнаруженные лица для последующей обработки.
- **stream_settings**: параметры видеопотока, дублирующие *параметры* в файле конфигурации `/etc/findface-video-manager.conf` (обладая при этом большим приоритетом).
- **status**: статус job-задания.
- **status_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.).
- **worker_id**: id экземпляра `findface-video-worker`, выполняющего job-задание.

`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, реализует HTTP- и веб-сокеты (вместе с Django), обновление базы данных и *вебхуки*.

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'
```

- 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?',
```

- Добавьте сервис `findface-facerouter` в автозагрузку Ubuntu и запустите сервис.

```
sudo systemctl enable findface-facerouter.service && sudo systemctl start findface-
↪facerouter.service
```

- 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:

- Поместите плагин в каталог по вашему выбору. Все используемые плагины должны находиться в одном каталоге.
- 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/'
```

- 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	Параметре from задано несуществующее лицо.
SOURCE_GALLERY_NOT_FOUND	Параметру 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**.

1.6.2 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](#).

1.6.3 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/

ant.v2.cpu.fnk      bee.v2.gpu.fnk      crop2x.v2_no_maxsize.cpu.fnk
ant.v2.gpu.fnk      crop1x.v2_maxsize400.cpu.fnk  crop2x.v2_no_maxsize.gpu.fnk
bee_fast.cpu.fnk    crop1x.v2_maxsize400.gpu.fnk  cropbbox.v2.cpu.fnk
bee_fast.gpu.fnk    crop2x.v2_maxsize400.cpu.fnk  cropbbox.v2.gpu.fnk
bee.v2.cpu.fnk      crop2x.v2_maxsize400.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.v1.cpu.fnk      gender.v2.cpu.fnk      medmask3.v2.cpu.fnk
age.v1.gpu.fnk      gender.v2.gpu.fnk      medmask3.v2.gpu.fnk
beard.v0.cpu.fnk    glasses3.v0.cpu.fnk    quality.v1.cpu.fnk
beard.v0.gpu.fnk    glasses3.v0.gpu.fnk    quality.v1.gpu.fnk
emotions.v1.cpu.fnk liveness.alley.v2.cpu.fnk
emotions.v1.gpu.fnk liveness.alley.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.v0.cpu.fnk  anaferon.v0.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.v0.cpu.fnk      carattr.quality.v0.cpu.fnk  
carattr.license_plate.v0.gpu.fnk      carattr.quality.v0.gpu.fnk  
carattr.license_plate_quality.v0.cpu.fnk  description.v0.cpu.fnk  
carattr.license_plate_quality.v0.gpu.fnk  description.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.color.v1.cpu.fnk  pedattr.quality.v0.cpu.fnk  
pedattr.color.v1.gpu.fnk  pedattr.quality.v0.gpu.fnk
```

1.6.4 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 идентификации объектов.
- Основная база данных системы на основе PostgreSQL, в которой хранятся внутренние данные системы, досье, учетные записи пользователей и настройки камер.
- Каталог `/var/lib/findface-security/uploads`, в котором хранятся загруженные в досье фотографии, видеофайлы, а также такие артефакты событий, как полные кадры, миниатюры объектов и нормализованные изображения объектов.
- Каталог `/var/lib/ffupload/`, в котором хранятся только такие артефакты событий, как миниатюры объектов.

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 силуэтов в фотографиях досье.
- `ffsec_car_events`: векторы признаков, извлеченные из обнаруженных на видео автомобилей.
- `ffsec_car_objects`: векторы признаков, извлеченные из автомобилей в фотографиях досье.
- `ffsec_face_events`: векторы признаков, извлеченные из обнаруженных на видео лиц.
- `ffsec_face_objects`: векторы признаков, извлеченные из лиц в фотографиях досье.
- `ffsec_user_face`: векторы признаков, извлеченные из фотографий пользователей FindFace Multi, используемые для аутентификации по лицу.
- `ffsec_persons`: центроиды персон (виртуальные векторы признаков, усредненные по всем лицам персоны) и метаданные.

1.6.5 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 хранилищ данных*.

```
findface-storage-api-dump --help

Usage of findface-storage-api-dump:
  -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-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-addr string
      Host:Port address (default "localhost:6379")
  -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-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
  -extraction-api-extraction-api string
      Extraction API address (default "http://127.0.0.1:18666")
  -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)
  -limits-allow-return-facen
      Allow returning raw feature vectors to detect responses if ?return_facen=true
  -limits-body-image-length int
```

(continues on next page)

(продолжение с предыдущей страницы)

```

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
    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-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-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")
-output-dir string
    Output directory (default ".")
-storage-api-max-idle-conns-per-host int
    storage-api-max-idle-conns-per-host (default 20)
-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)

```

1.6.6 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 хранилищ данных*.

```
findface-storage-api-restore --help
```

Usage of findface-storage-api-restore:

```
-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-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-addr string
    Host:Port address (default "localhost:6379")
-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-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
-cpu-profile string
    Enable CPU profile and set output file
-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-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)
-limits-allow-return-facen
    Allow returning raw feature vectors to detect responses if ?return_facen=true
-limits-body-image-length int
```

(continues on next page)

(продолжение с предыдущей страницы)

```

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
    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-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-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")
-rename string
    Ignore dump header and use this string as gallery name
-storage-api-max-idle-conns-per-host int
    storage-api-max-idle-conns-per-host (default 20)
-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)

```


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.

2.1 Первые шаги после установки

После успешного развертывания FindFace Multi пришло время *выполнить первичную настройку*, 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*
- *Создание пользователей и обеспечение безопасности системы*
- *Добавление камер*
- *Составление списков наблюдения и загрузка досье*
- *Мониторинг объектов*
- *Организация видеонаблюдения*
- *Подсчет лиц и силуэтов*
- *Анализ людей*
- *Работа в штатном режиме*
- *Базовое обслуживание системы*

- *Большие возможностей*

2.1.1 Подготовка к работе

Выполните первичную настройку системы:

1. *Настройте* левую панель навигации.
2. *Адаптируйте* основные настройки.
3. *Выберите* язык.

Дополнительно:

1. *Включите* распознавание атрибутов лиц
2. *Защитите* систему от спуфинга, включив функцию распознавания живых лиц (*Liveness*)
3. *Включите* распознавание автомобилей и атрибутов автомобилей
4. *Включите* распознавание силуэтов и атрибутов силуэтов

2.1.2 Создание пользователей и обеспечение безопасности системы

1. Ознакомьтесь со списком *предустановленных пользовательских ролей* и при необходимости *создайте новые*.
2. *Добавьте пользователей* в систему и предоставьте им полномочия.
3. *Настройте* аутентификацию и мониторинг сессии пользователя. Аутентификация возможна по паролю, лицу, лицу или паролю, лицу и паролю.

Дополнительно:

1. *Включите* SSL-шифрование данных.
2. *Включите* защиту досье. Если защита досье отключена, фотографии и вложения досье будут доступны по прямой ссылке независимо от прав пользователя.
3. *Отключите* ACL FindFace Multi, если не нуждаетесь в данной функции, поскольку постоянные проверки разрешений потребляют большой объем системных ресурсов.

2.1.3 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.*

2.1.4 Составление списков наблюдения и загрузка досье

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 досье и добавьте их в созданный список наблюдения *вручную, пакетно через веб-интерфейс* или используя функцию *пакетной загрузки через консоль*.

Дополнительно:

1. *Распределите базу данных досье* между несколькими серверами. База данных досье будет доступна для редактирования на головном сервере, а для чтения и мониторинга — на ведомых.
2. *Настройте содержимое досье.* Создайте дополнительные поля, вкладки и поисковые фильтры.

2.1.5 Мониторинг объектов

По умолчанию FindFace Multi отслеживает только незарегистрированные в системе объекты, т. е. объекты, для которых *отсутствуют совпадения с досье*. 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 поддержку законов, связанных с обработкой персональных данных (GDPR и аналогичные). *Подробнее.*

2.1.6 Организация видеонаблюдения

Настройте раскладку камер для базового видеонаблюдения.

2.1.7 Подсчет лиц и силуэтов

Настройте *счетчики* для подсчета лиц и силуэтов 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 и т. д.

2.1.8 Анализ людей

FindFace Multi предоставляет набор инструментов для анализа людей:

1. Включите *распознавание персон*, чтобы построить галерею персон. В системных базах данных появится новая сущность «person event» (событие появления персоны), связанная со всеми *эпизодами*, в которых фигурировало лицо того или иного человека. Вы можете работать с галереей персон по аналогии со списками событий и эпизодов.
2. *Анализируйте* социальные взаимодействия. Изучите круг людей, с которыми человек контактировал.
3. 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.*

2.1.9 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* (только для событий распознавания лиц). Эпизод — это набор событий идентификации, в которых фигурируют лица одного и того же человека, 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 лиц. Это позволяет с легкостью обрабатывать разнородные события даже в больших количествах.
3. Ищите объекты в базе обнаруженных объектов и базе досье. *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, персоны, камеры, досье, аналитику KYC, журнал действий пользователей.

2.1.10 Базовое обслуживание системы

1. *Настройте* автоматическое удаление старых событий, эпизодов и полных кадров.
2. При необходимости *удалите* старые события, эпизоды и полные кадры вручную.
3. Регулярно создавайте *резервную копию* базы данных.
4. Поддерживайте безопасность системы, *анализируя журнал действий пользователей*.

2.1.11 Больше возможностей

1. Настройте *вебхуки* для автоматической отправки уведомлений об определенных событиях, эпизодах и записях счетчика на заданный URL-адрес. При наступлении нужного события FindFace Multi отправит HTTP-запрос на URL-адрес, указанный в настройках вебхука. Вебхуки можно использовать для решения разнообразных задач, например, для уведомления пользователя об определенном событии, вызова определенных действий на целевом веб-сайте, при решении задач безопасности, таких как удаленное автоматическое управление доступом и др. *Подробнее*.
2. Задействуйте функции FindFace Multi через *HTTP API*.
3. Ознакомьтесь со списком наших *партнерских интеграций*.

См.также:

- *Первичная настройка*
- *Управление пользователями и безопасность системы*
- *Управление видеокамерами*
- *Настройка мониторинга объектов. База данных досъе*
- *Идентификация объектов в офлайн видео*
- *Счетчики лиц и силуэтов*
- *Распознавание персон и связанная с людьми аналитика*
- *Расширенный функционал*
- *Обслуживание и устранение неисправностей*

2.2 Работа с FindFace Multi

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*.

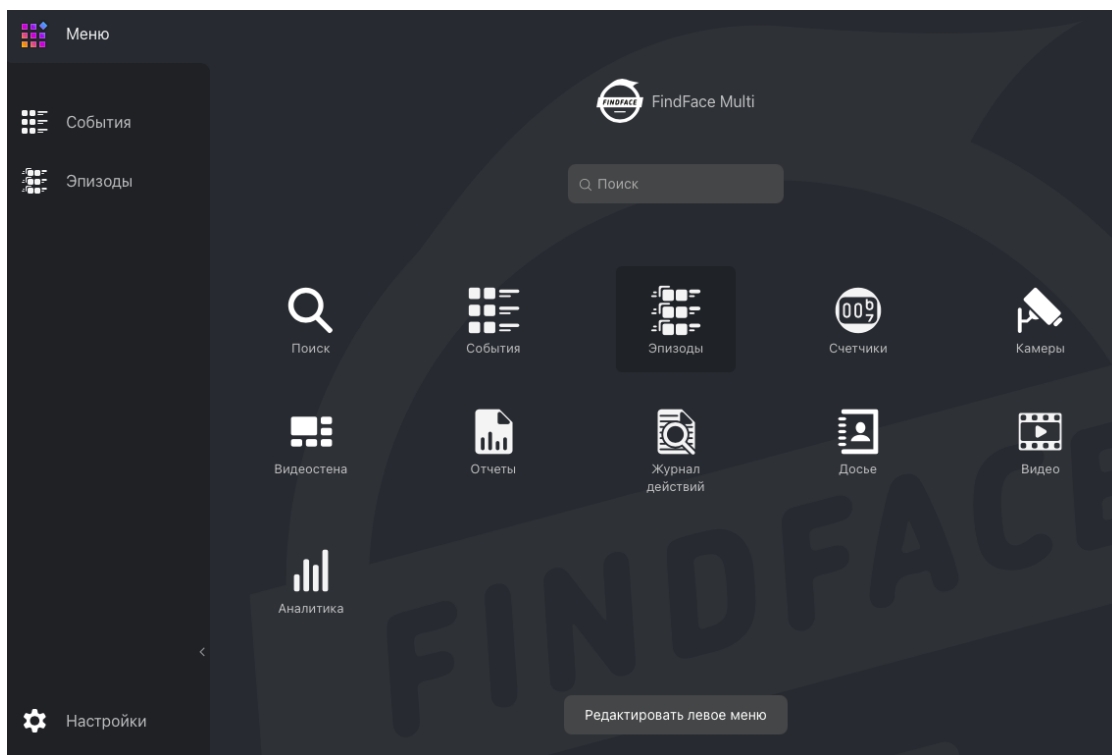
Совет: Поднимите безопасность своей системы на новый уровень, используя *аутентификацию по лицу*.

2.2.1 Первичная настройка

Этот раздел посвящен первичной настройке FindFace Multi. Узнайте, как настроить левую панель навигации, пороги распознавания объектов, запланировать автоматическую очистку базы данных, переключить язык и многое другое.

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*.

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 видеодетекторе 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.v1.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.v1.cpu.fnk
	GPU	face_age: faceattr/age.v1.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.v1.cpu.fnk  age.v1.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.alley.
↪v2.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 = {
    ...
    'EVENTS_FEATURES': ['gender', 'age', 'emotions', 'beard', 'glasses', 'medmask']
    ...
}
```

6. Перезапустите `findface-security`.

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

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 анти-спуфинговую 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`) и нормализатору (`norm`), которые будут 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.alleyn.v2.gpu.fnk

## path to normalization for liveness
## type:string env:CFG_LIVENESS_NORM longopt:--liveness-norm
norm = /usr/share/findface-data/models/facenorm/crop2x.v2_maxsize400.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.alleyn.v2.cpu.fnk

## path to normalization for liveness
## type:string env:CFG_LIVENESS_NORM longopt:--liveness-norm
norm = /usr/share/findface-data/models/facenorm/crop2x.v2_maxsize400.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.

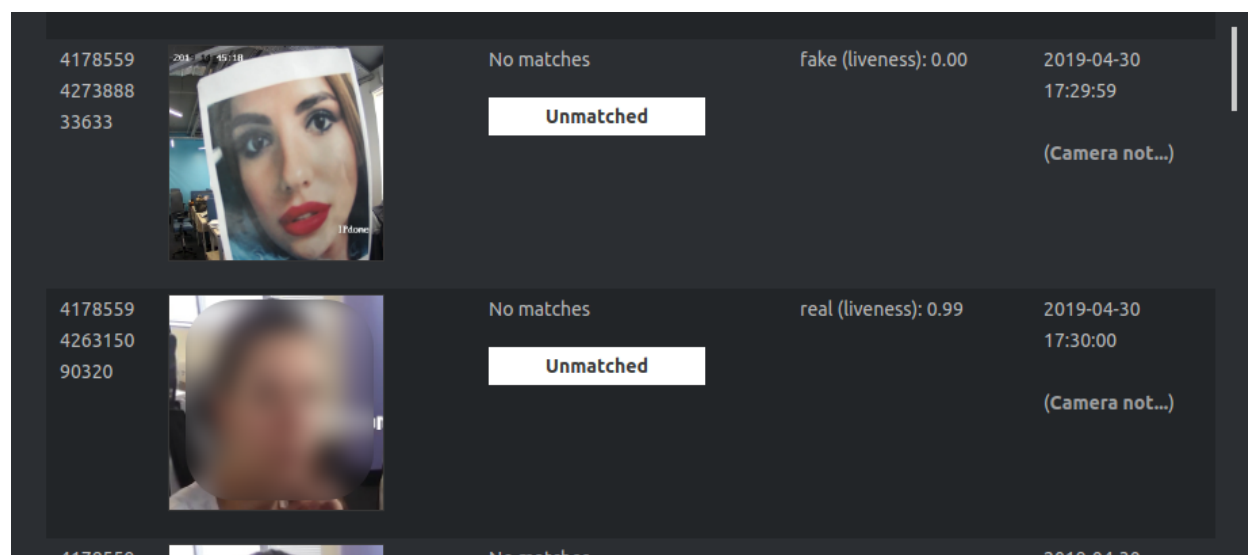
Примечание: Значение по умолчанию является оптимальным. Перед изменением порога проконсультируйтесь у наших специалистов по адресу support@ntechlab.com.

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

'LIVENESS_THRESHOLD': 0.85,
```

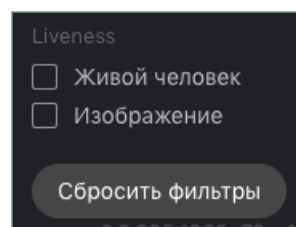
Информация о живом лице в веб-интерфейсе

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

Распознавание автомобилей и их атрибутов

FindFace Multi позволяет распознавать отдельные автомобили и их атрибуты.

Атрибуты следующие:

- номерной знак (для отдельных стран),
- цвет,
- марка,
- модель,
- тип кузова.

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 и модель. 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 пошаговых инструкциях 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.v0.gpu.fnk
        car_license_plate_quality: carattr/carattr.license_plate_quality.v0.gpu.fnk
        car_make: ''
        car_quality: carattr/carattr.quality.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.v0.cpu.fnk
        car_license_plate_quality: carattr/carattr.license_plate_quality.v0.cpu.fnk
        car_make: ''
        car_quality: carattr/carattr.quality.v0.cpu.fnk
```

Доступны следующие экстракторы:

Экстрактор	Настройка
отдельный объект типа автомобиль	car_emben: carrec/alonso.cpu.fnk
	car_emben: carrec/alonso.gpu.fnk
номерной знак	car_license_plate: carattr/carattr.license_plate.v0.cpu.fnk car_license_plate_quality: carattr/carattr.license_plate_quality.v0.cpu.fnk
	car_license_plate: carattr/carattr.license_plate.v0.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

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: ''
        car_license_plate: ''
        car_license_plate_quality: ''
        car_make: ''
        car_quality: ''
        car_trash: ''
```

- 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.v0.gpu.fnk carnorm/anaferon.v0.cpu.fnk	car_license_plate
cropbbox	facenorm/cropbbox.v2.gpu.fnk facenorm/cropbbox.v2.cpu.fnk	car_license_plate_quality, car_description, car_quality

GPU

```
normalizers:
    ...

models:
    carlicplate:
        model: carnorm/anaferon.v0.gpu.fnk
    ...
    cropbbox:
        model: facenorm/cropbbox.v2.gpu.fnk
```

CPU

```
normalizers:
    ...

models:
    carlicplate:
        model: carnorm/anaferon.v0.cpu.fnk
    ...
    cropbbox:
        model: facenorm/cropbbox.v2.cpu.fnk
```

5. Убедитесь, что раздел `objects -> car` содержит `quality_attribute: car_quality` и `base_normalizer: facenorm/cropbbox.v2.*.fnk`:

GPU

```
objects:
    ...
    car:
        base_normalizer: facenorm/cropbbox.v2.gpu.fnk
        quality_attribute: car_quality
    ...
```


CPU

```
objects:
  ...
  car:
    base_normalizer: facenorm/cropbbox.v2.cpu.fnk
    quality_attribute: car_quality
```

6. После внесения изменений в файл конфигурации убедитесь, что он выглядит аналогично следующему примеру.
7. 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
## type:string env:CFG_CAR_NORM_QUALITY longopt:--car-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

#-----
[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: 0.73
    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

```

(continues on next page)

(продолжение с предыдущей страницы)

```

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

```

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
'CAR_EVENTS_FEATURES': ['description', 'license_plate'],

```

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.

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},

```

(continues on next page)

(продолжение с предыдущей страницы)

```
'model': {'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
```

Распознавание силуэтов и их атрибутов

FindFace Multi позволяет распознавать отдельные человеческие силуэты и атрибуты силуэтов (в текущей реализации только цвет одежды).

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_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_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
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_color: ''
        body_emben: ''
        body_quality: ''
```

4. 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
```

5. 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. После внесения изменений в файл конфигурации убедитесь, что он выглядит аналогично следующему примеру.
7. 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
```

(continues on next page)

(продолжение с предыдущей страницы)

```

## 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

```

```
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:
    ...
    body:
        filter_min_quality: 0.6
        filter_min_size: 1
        filter_max_size: 8192
        roi: ''
        fullframe_crop_rot: false
        fullframe_use_png: false
        jpeg_quality: 95
        overall_only: true
        realtime_post_first_immediately: false
        realtime_post_interval: 1
        realtime_post_every_interval: false
        track_interpolate_bboxes: true
        track_miss_interval: 1
        track_overlap_threshold: 0.25
        track_max_duration_frames: 0
        track_send_history: false
        post_best_track_frame: true
        post_best_track_normalize: true
        post_first_track_frame: false
        post_last_track_frame: false
```

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 (цвет одежды), которые требуется отображать в событиях распознавания.

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

3. Перезапустите сервис `findface-security`.

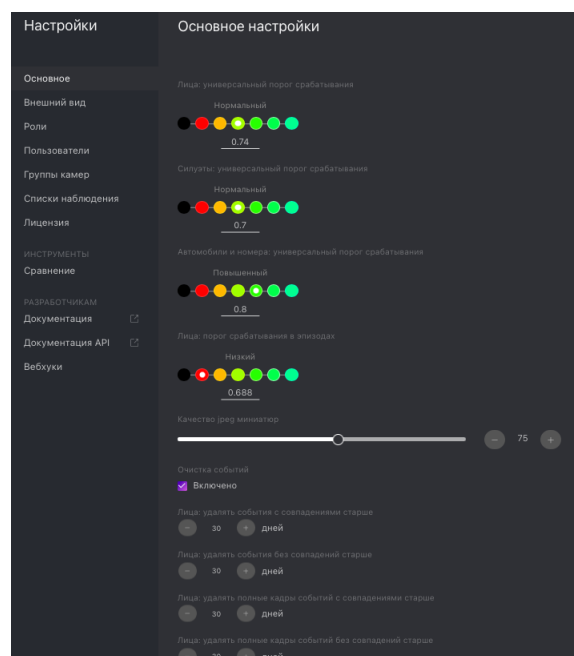
```
sudo systemctl restart findface-security.service
```

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)
- порог срабатывания для эпизодов распознавания лиц
- 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*
- *Порог срабатывания для эпизодов (только лица)*
- *Качество JPEG миниатюр*
- *Автоматическое удаление старых событий и эпизодов*

Универсальный порог срабатывания

FindFace Multi принимает решение о совпадении обнаруженного лица с лицом из досье (т. е. о положительной верификации) 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 данные объекты добавлены в систему. На основании предустановленного порога FindFace Multi принимает решение о принадлежности одному человеку обнаруженного силуэта и силуэта из досье. Аналогично, обнаруженный автомобиль будет считаться совпадающим с автомобилем из досье.

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

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

Порог срабатывания для эпизодов (только лица)

Примечание: В текущей реализации FindFace Multi поддерживает только эпизоды распознавания лиц. Эпизоды распознавания автомобилей и силуэтов будут добавлены в следующих версиях.

При формировании эпизода система ищет в базе данных векторов признаков *недавние* события с похожими лицами выше определенной пороговой степени схожести. Порог по умолчанию установлен на оптимальное значение. При необходимости вы можете изменить его. Обязательно свяжитесь с нашими экспертами для консультации (support@ntechlab.com).

Качество JPEG миниатюр

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

Автоматическое удаление старых событий и эпизодов

Используйте эту же вкладку, чтобы задать расписание автоматического удаления старых событий и эпизодов из базы данных. Можно настроить удаление событий и эпизодов по разным расписаниям в зависимости от наличия совпадений с базой данных досье, а также выбрать удаление только полных кадров. Вы также можете по отдельности удалять события с лицами, автомобилями и силуэтами.

Язык веб-интерфейса

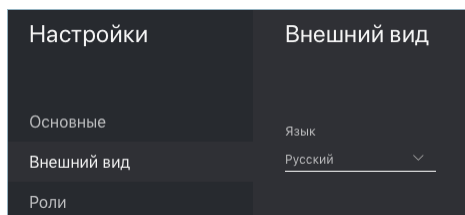
По умолчанию в веб-интерфейсе для выбора доступны только английский и русский языки. Другие языки предоставляются по запросу (support@ntechlab.com) и добавляются в систему через файл конфигурации `/etc/findface-security/config.py`.

В этом разделе:

- Переключение языка в веб-интерфейсе
- Добавление пользовательского языка

Переключение языка в веб-интерфейсе

Для того чтобы переключить язык системы, перейдите в меню *Настройки* -> *Внешний вид*.



Добавление пользовательского языка

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": [  
            {
```

(continues on next page)

(продолжение с предыдущей страницы)

```

    "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.

2.2.2 Управление пользователями и безопасность системы

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

Управление 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. Обладает полными правами на *управление видеокамерами*, *базой данных досье*, *событий*, пользователями FindFace Multi, а также полным доступом ко всем остальным функциям.

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

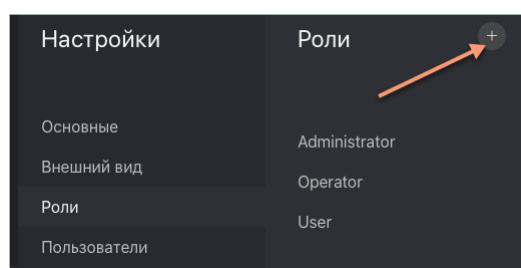
- Оператор может создавать досье вручную, получать и подтверждать события, выполнять поиск объектов в списке событий. Остальные данные доступны в режиме чтения. Возможность *пакетного создания досье* недоступна.
- Пользователь. Обладает правами только на подтверждение событий и поиск объектов в базе событий и досье. Остальная информация доступна в режиме чтения.

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

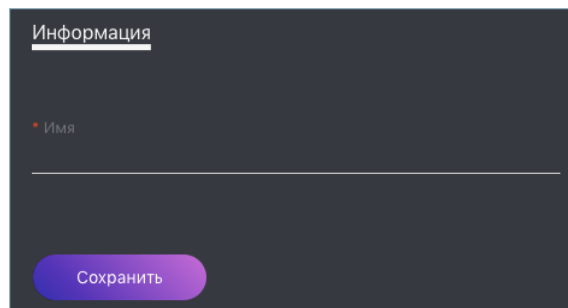
Создание новой роли

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 досье, требуются только права на камеру.

Полный список сущностей FindFace Multi:

- faceevent: *событие распознавания лица*
- faceobject: *фотография лица в досье*
- carevent: *событие распознавания автомобиля*
- carobject: *фотография автомобиля в досье*
- bodyevent: *событие распознавания силуэта*
- bodyobject: *фотография в полный рост в досье*
- dossierlist: *списки наблюдения*
- dossier: *досье*
- cameragroup: *группа камер*
- camera: *камера*
- eventepisode: *эпизоды*
- person: *галерея персон*
- 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*

Вы также можете включать и отключать права на следующий функционал:

- configure_ntls: *конфигурация сервера лицензий findface-ntls*
- batchupload_dossier: *пакетное создание досье*
- view_runtimeetting: *просмотр основных настроек FindFace Multi*
- change_runtimeetting: *изменение основных настроек FindFace Multi*
- view_auditlog: *просмотр и работа с аудитлогами.*

Информация

Списки наблюдения

Группы камер

Разрешения

Выбрать все

Отменить все

Имя	Просмотр	Изменить	Добавить	Удалить
faceevent				
faceobject				
carevent				
carobject				
bodyevent				
bodyobject				
dossierlist				
dossier				
cameragroup				
camera				
eventepisode				
person				
uploadlist				
upload				
user				
webhook				
videoarchive				
counter				
metadictionary				
notification				
report				
Имя	Активный			
configure_ntls				
batchupload_dossier				
view_runtimesetting				
change_runtimesetting				
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 досье в этих списках), созданным пользователем с той же основной ролью, с учетом привилегий, определенных их дополнительными ролями.

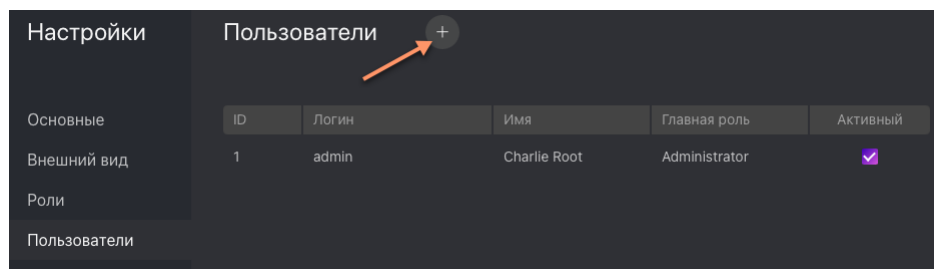
См.также:

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 они 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*.

Предоставление администратору прав на системные плагины

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

- *интеграции с партнерами*,
- *управление распределенной базой досье*,
- 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
```

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.

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. 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/{}'
```

Включение защиты досье

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

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

Для включения защиты досье 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
```

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
```

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

« < Страница 1 > »

Пользователь	IP	ID устройства	Действие	Объект	ID объекта	Время
1	172.20.78.26	585607ef-3551-4eb2-a54f-1bca5568617f	Поиск	Досье		2021-06-30 21:50:27
1	172.20.78.26	585607ef-3551-4eb2-a54f-1bca5568617f	Создать	Лицо из досье	436223953190142176	2021-06-30 21:47:41
1	172.20.78.26	585607ef-3551-4eb2-a54f-1bca5568617f	Создать	Досье	4	2021-06-30 21:47:40
1	172.20.78.26	585607ef-3551-4eb2-a54f-1bca5568617f	Поиск	Досье		2021-06-30 21:47:13
1	172.20.78.26	585607ef-3551-4eb2-a54f-1bca5568617f	Авторизация	Пользователь	1	2021-06-30 21:45:33
1	172.20.128.18	25ffc88b-ab4b-4703-a8a1-b399fd6609fe	Авторизация	Пользователь	1	2021-06-30 21:16:46
1	172.20.178.58	891f05e7-db8a-4b54-b752-430c4824fd48	Подтверждение	Событие с лицом		2021-06-30 20:54:44
1	172.20.178.58	891f05e7-db8a-4b54-b752-430c4824fd48	Подтверждение	Событие с лицом		2021-06-30 20:15:30
1	172.20.178.58	891f05e7-db8a-4b54-b752-430c4824fd48	Создать	Лицо из досье	4362224625955010716	2021-06-30 20:15:07
1	172.20.178.58	891f05e7-db8a-4b54-b752-430c4824fd48	Поиск	Досье		2021-06-30 20:15:05
1	172.20.178.58	891f05e7-db8a-4b54-b752-430c4824fd48	Подтверждение	Событие с лицом		2021-06-30 20:14:28
1	172.20.178.58	891f05e7-db8a-4b54-b752-430c4824fd48	Подтверждение	Событие с лицом		2021-06-30 19:47:10
1	172.20.178.58	891f05e7-db8a-4b54-b752-430c4824fd48	Редактировать	Событие с лицом	4362219947482345189	2021-06-30 19:46:49
1	172.20.178.58	891f05e7-db8a-4b54-b752-430c4824fd48	Создать	Отчет	2	2021-06-30 19:32:14
1	172.20.178.58	891f05e7-db8a-4b54-b752-430c4824fd48	Редактировать	Событие с лицом	4362217392464108924	2021-06-30 19:30:58
1	172.20.178.58	891f05e7-db8a-4b54-b752-430c4824fd48	Редактировать	Список наблюдения	2	2021-06-30 19:29:01

Старт

Конец

IP

ID устройства

Объект

Не выбраны

ID объекта

Действие

Не выбраны

Пользователь

Сбросить фильтры

Создать отчет

Каждая запись журнала содержит следующие данные:

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

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

2.2.3 Управление видеокамерами

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:

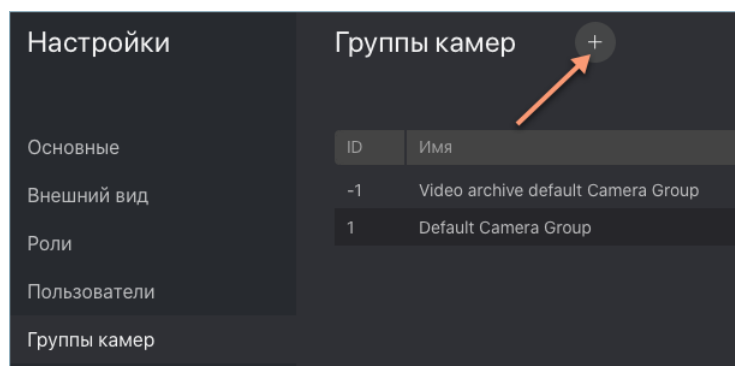
- *Создание группы камер*
- *Добавление камеры*
- *Мониторинг работы камер*

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.

The screenshot shows the 'Создать группу камер' (Create Camera Group) form, 'Информация' (Information) tab. The form contains the following fields and options:

- Имя** (Name): Required field, currently empty.
- Офис** (Office): Text input field.
- Комментарий** (Comment): Text input field.
- Метки** (Tags): Dropdown menu with the text 'Введите или выберите метки' (Enter or select tags).
- Дедуплицировать события** (Deduplicate events): Checked checkbox.
- Интервал дедупликации** (Deduplication interval): Required field, currently set to 15.
- Порог срабатывания** (Trigger threshold): Unchecked checkbox.
- Активная** (Active): Checked checkbox.

Buttons at the bottom: 'Сохранить' (Save) and 'Назад' (Back).

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

Примечание: Для того чтобы завершить выделение, перечислите метки в файле конфигурации `/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 проконсультируйтесь с нашими техническими экспертами по адресу 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.

Имя	Отсутств...	Просмотр	Изменить
Administrator	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Operator	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
User	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>

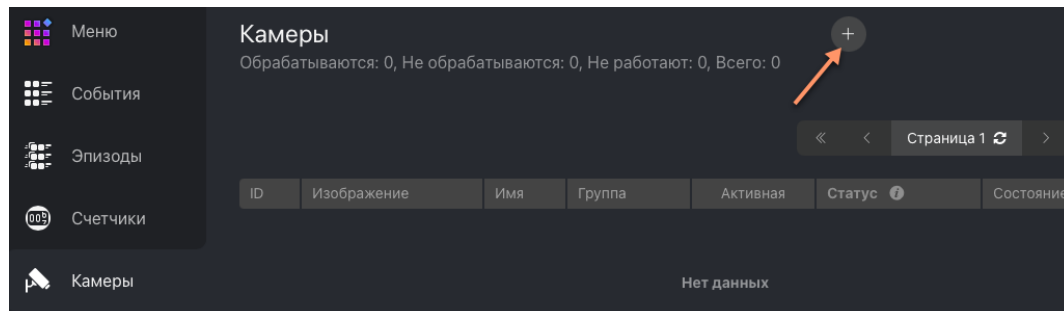
Сохранить Назад

- Нажмите *Сохранить*.

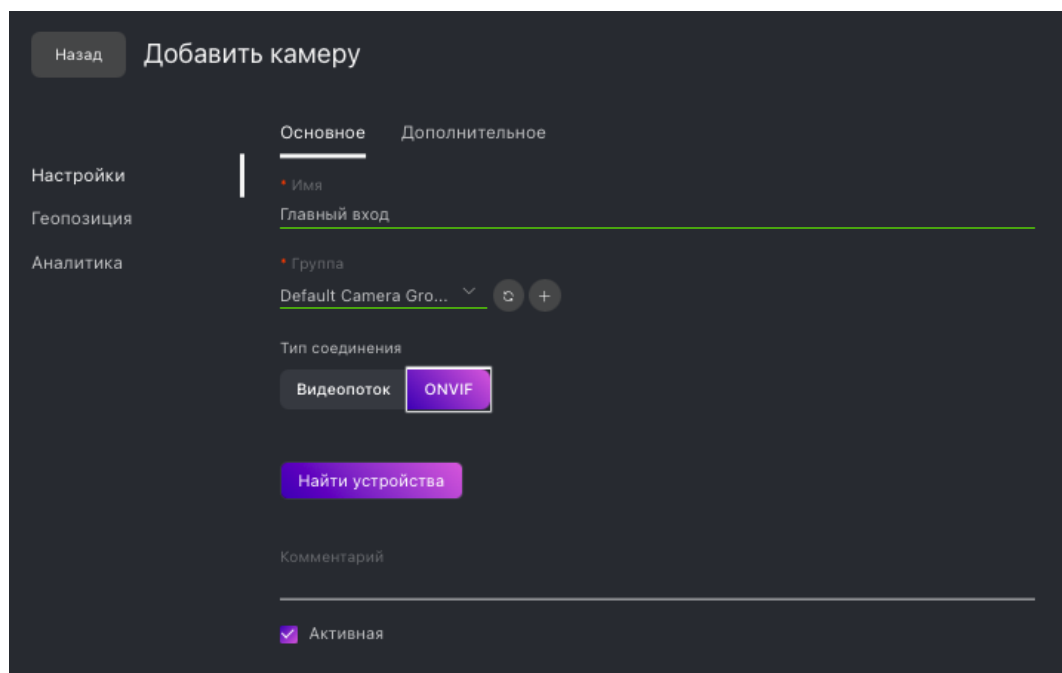
Добавление камеры

Для добавления камеры в группу выполните следующие действия:

1. Перейдите на вкладку *Камеры*.
2. Нажмите **+**.



3. 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.
- 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 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.

- *Проверять 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.

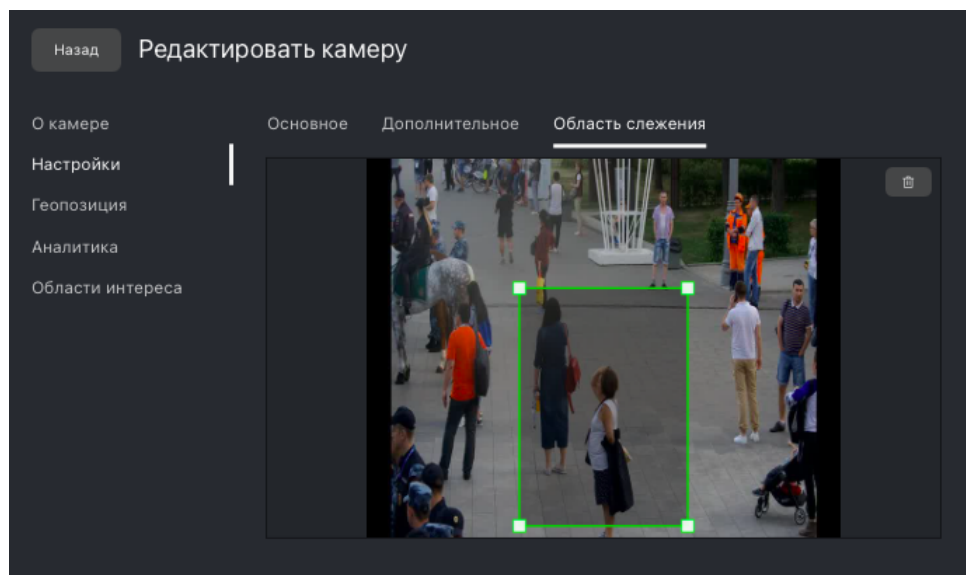
- *Минимальное качество изображения объекта*: Минимальное качество изображения объекта для отправки на сервер. Не меняйте значение по умолчанию без предварительной консультации с нашими техническими специалистами (support@ntechlab.com).
- *Минимальный размер объекта*: Минимальный размер объекта в пикселях для отправки на

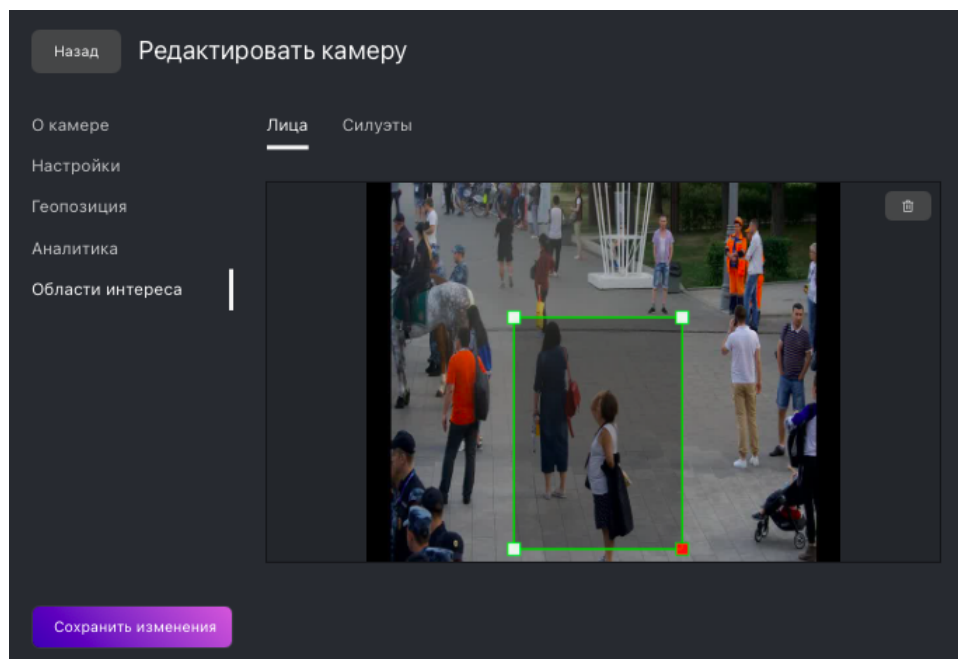
сервер.

- *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 изображение 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*.

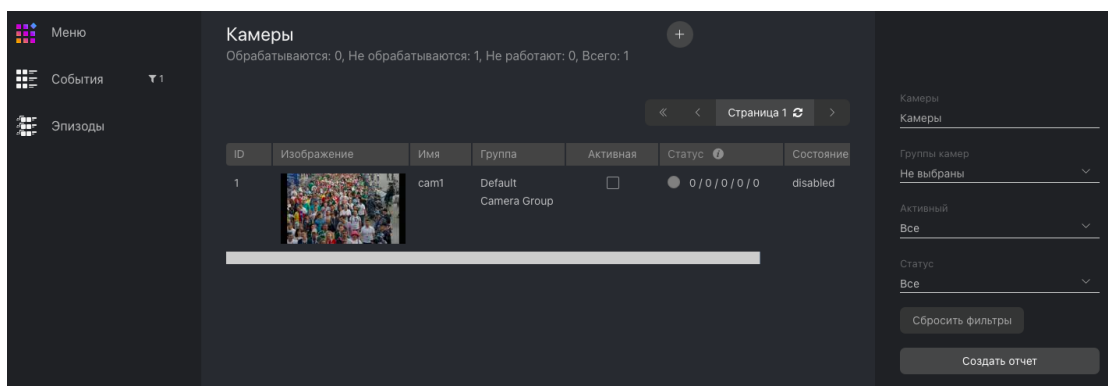




Примечание: Каждая камера связана с так называемым job-заданием, задачей на обработку видео, содержащей параметры конфигурации и данные видеопотока, которая назначается определенному экземпляру `findface-video-worker`. Данная задача может быть перезапущена (см. [Мониторинг работы камер](#)).

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},
    ...
}
```

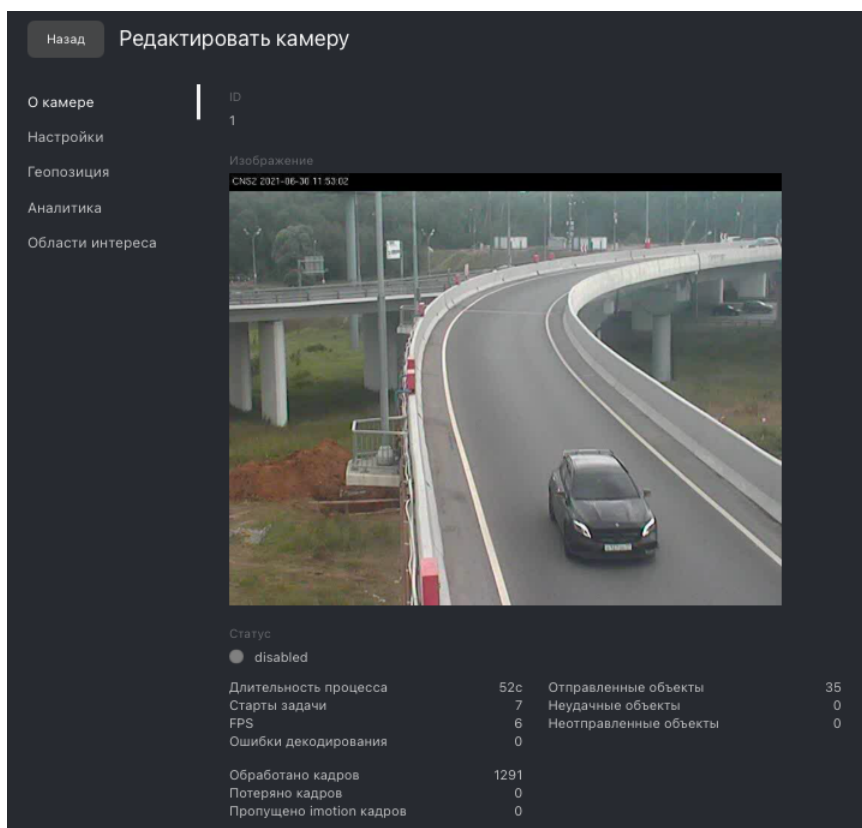
Каждая камера связана с так называемым job-заданием, задачей на обработку видео, содержащей параметры конфигурации и данные видеопотока, которая назначается определенному экземпляру `findface-video-worker`. Данная задача может быть перезапущена.

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*.



См.также:

- *Привязка группы камер к экземпляру findface-video-worker*
- *Дедупликация событий*
- *Видеодетекция объектов: findface-video-manager и findface-video-worker*

2.2.4 Настройка мониторинга объектов. База данных досье

Эта глава посвящена настройке мониторинга объектов и созданию базы данных досье.

FindFace Multi обеспечивает видеомониторинг следующих объектов:

- человеческое лицо
- человеческий силуэт
- автомобиль

Мониторинг объектов реализован посредством совокупности встроенных и пользовательских списков наблюдения, например, Разыскивается, VIP и т. п., и базы данных досье. Вы можете создать столько пользовательских списков наблюдения, сколько необходимо. Каждое досье содержит одну или несколько фотографий объекта и относится к одному или нескольким спискам наблюдения. Для того чтобы поставить объект на мониторинг, необходимо сделать активным соответствующий список наблюдения.

Вы можете сохранять смешанные данные и объекты разных типов в одном досье, если данные объекты связаны. Например, досье может содержать лицо и фотографию в полный рост, фотографию автомобиля и номерной знак, характеризующие разыскиваемого преступника. В этом случае система будет искать любой из указанных объектов в поле зрения камеры.

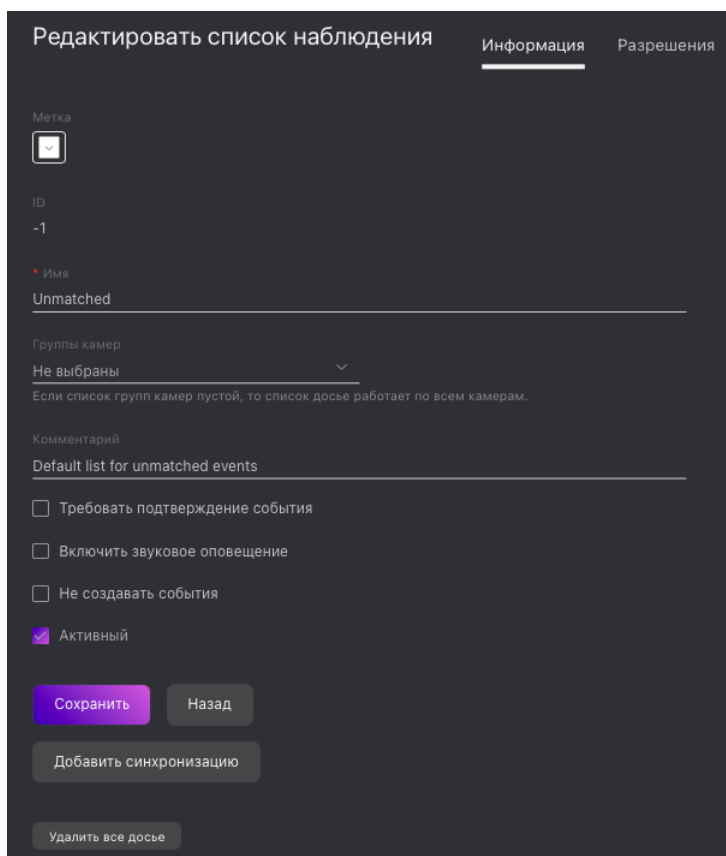
Совет: Для автоматического создания большого количества досье используйте *функционал пакетной загрузки*.

В этом разделе:

- *Мониторинг незарегистрированных объектов*
- *Создание списка наблюдения*
- *Создание досье вручную*
- *Пакетная загрузка фотографий*
- *Фильтры досье*
- *Очистка базы данных досье*
- *Отключение создания событий для определенных объектов*

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 досье. 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.

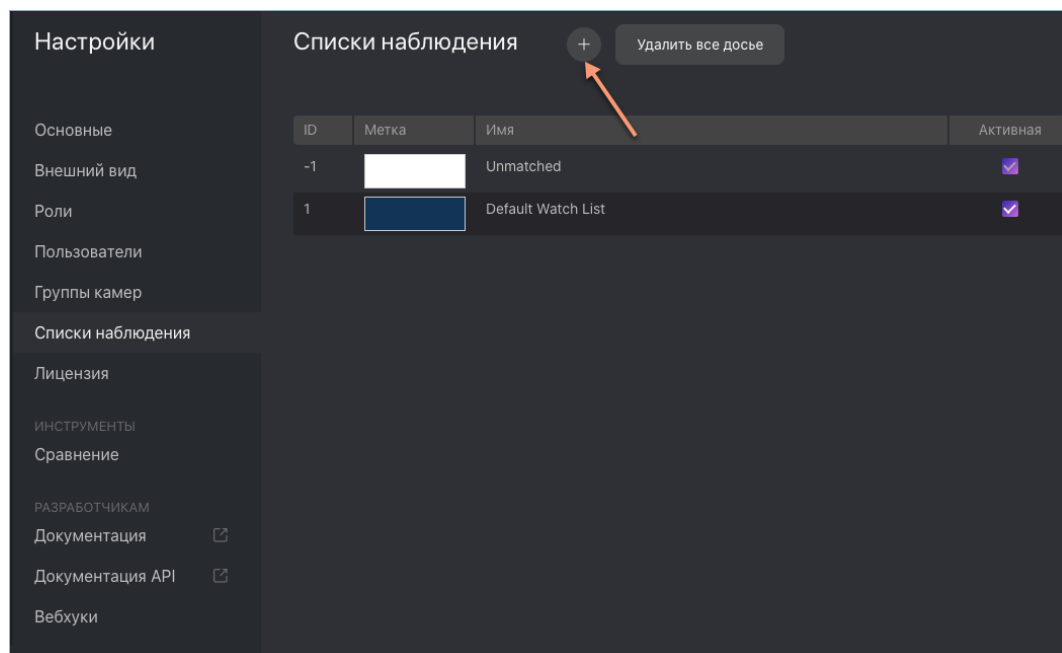


Примечание: Для того чтобы отобразить в списке событий только незарегистрированные объекты, выберите значение *Без совпадений* в фильтре *Совпадения* на вкладке *События*.

Создание списка наблюдения

Для создания пользовательского списка наблюдения выполните следующие действия:

1. Перейдите на вкладку *Настройки*. Выберите *Списки наблюдения*.
2. Нажмите *+*.



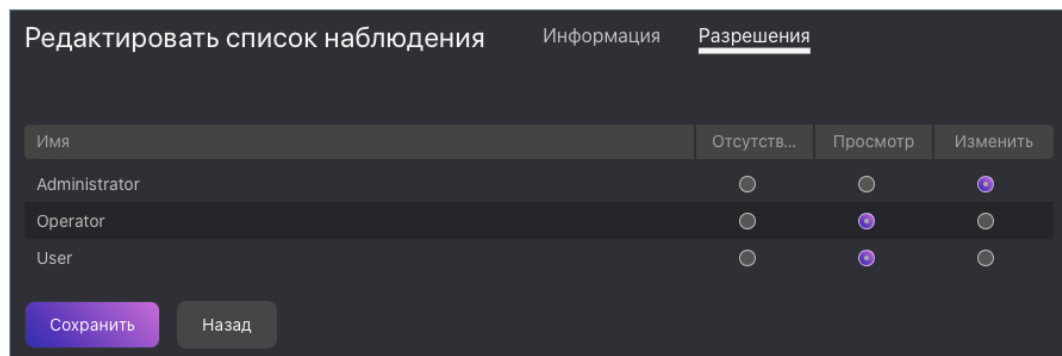
- В палитре *Метка* выберите цвет, который будет использоваться в оповещениях о событиях распознавания из данного списка. Правильно выбранный цвет повышает быстроту реагирования оператора на событие.

- 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 проконсультируйтесь с нашими техническими экспертами по адресу 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.

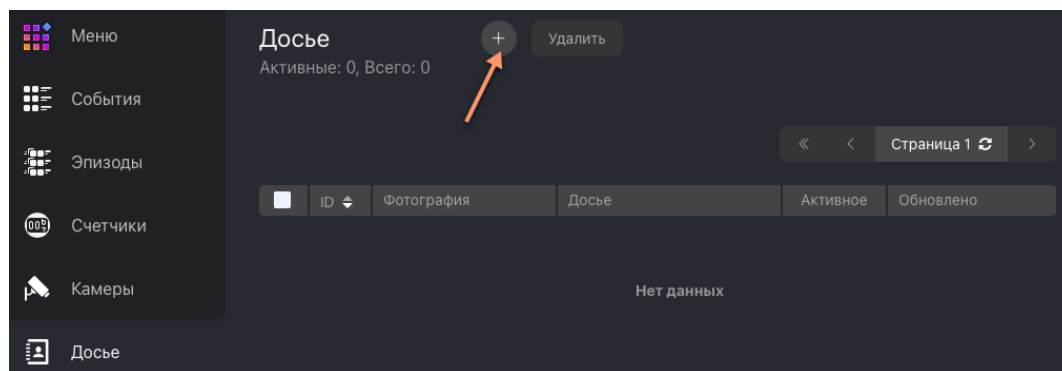


12. Нажмите *Сохранить*.

Создание досье вручную

Для создания досье вручную выполните следующие действия:

1. В веб-интерфейсе перейдите на вкладку *Досье*.
2. Нажмите *+*.



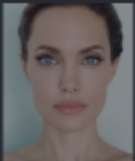

3. Добавьте фотографии по крайней мере одного из следующих объектов: лицо, силуэт, автомобиль. Поддерживаемые форматы: WEBP, JPG, BMP, PNG.

Важно: Лицо или силуэт на фотографии должны быть надлежащего качества, т. е. в близком к фронтальному положении. Расстояние между зрачками: 60 px. При несоответствии фотографии требованиям будет выведено сообщение с описанием ошибки.


Создать досье
Много досье для загрузки? Попробуйте [Пакетную загрузку досье](#)

Фотографии



Лица


Силуэты



Автомобили и номера




Файлы



* Имя
Илианна Скотт

Комментарий

Регистрационный номер

* Списки наблюдения
Default Watch List   

☒ Активное

4. Прикрепите имеющиеся отношение к делу файлы.
5. Введите имя человека. При необходимости добавьте комментарий.
6. При необходимости укажите номер автомобиля.
7. Из раскрывающегося списка *Списки наблюдения* выберите список (или несколько списков, по очереди), в который следует добавить досье.
8. Поставьте флажок *Активное*. Если досье неактивно, оно не будет использоваться для идентификации объектов в режиме реального времени.

9. Нажмите *Сохранить*. Если похожее досье уже существует в базе данных, вы сможете выбрать, объединить ли его с новым досье, отменить создание нового досье, создать новое.

Пакетная загрузка фотографий

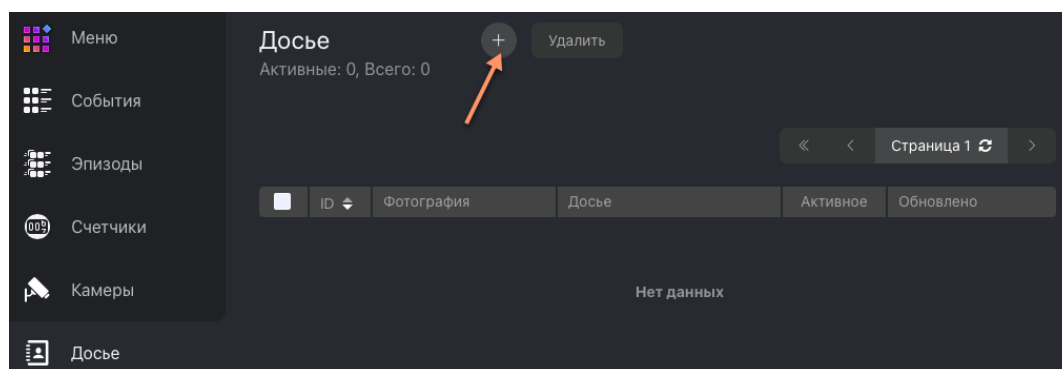
Для создания большого количества досье используйте функционал пакетной загрузки. Выполните следующие действия:

Совет: Если количество досье превышает 10000, используйте *Пакетная загрузка фотографий через консоль*.

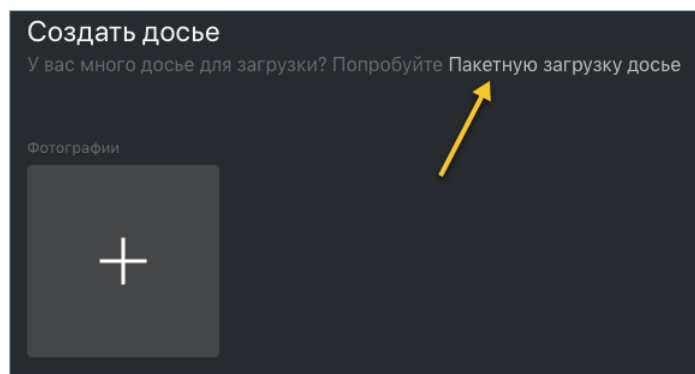
Важно: Поддерживаемые форматы: WEBP, JPG, BMP, PNG.

Важно: Лицо или силуэт на фотографии должны быть надлежащего качества, т. е. в близком к фронтальному положении. Расстояние между зрачками: 60 пикселей. При несоответствии фотографии требованиям будет выведено сообщение с описанием ошибки.

1. В веб-интерфейсе перейдите на вкладку *Досье*.
2. Нажмите *+*.

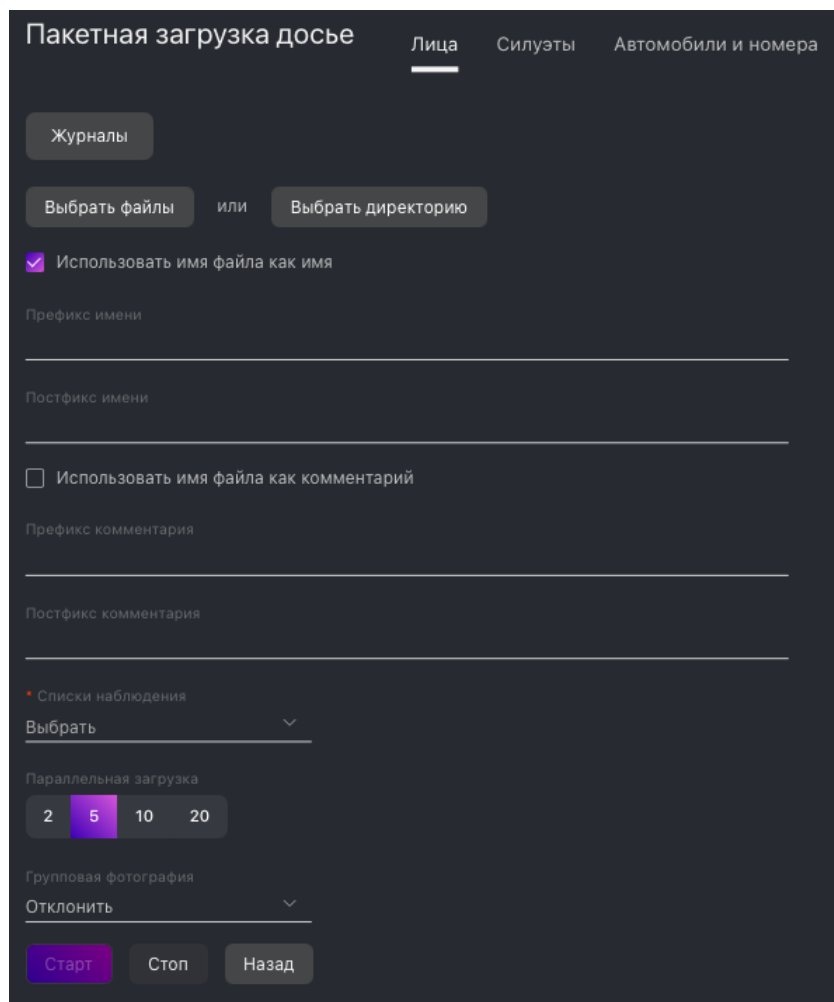


3. Нажмите *Пакетная загрузка досье*.



4. Укажите тип объектов для обнаружения на фотографиях.

5. Выберите фотографии для загрузки пофайлово или укажите каталог с фотографиями.



6. Имена файлов с фотографиями можно использовать как основу для имен и/или комментариев в создаваемых досье. 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.

7. Из раскрывающегося списка *Списки наблюдения* выберите список (или несколько списков, по очереди), в который следует добавить создаваемые досье.
8. В параметре *Параллельная загрузка* задайте количество потоков загрузки фотографий. Чем больше потоков, тем быстрее будет завершена загрузка, однако также потребуется и большее количество ресурсов.
9. 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.
10. Для запуска пакетного создания досье 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



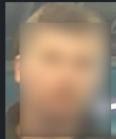
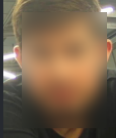
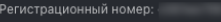

<< < Страница 1 > >>

Фильтры досье

Все созданные в FindFace Multi досье отображаются на вкладке *Досье*. Для сортировки досье используйте фильтры.

Досье + Удалить

Активные: 3, Всего: 3 << < Страница 1 > >>

<input type="checkbox"/>	ID	Фотография	Досье	Активное	Обновлено
<input type="checkbox"/>	5		NTL team Персона 31, 	<input checked="" type="checkbox"/>	2021-07-02 16:05:43
<input type="checkbox"/>	3		NTL team	<input checked="" type="checkbox"/>	2021-07-07 17:07:26
<input type="checkbox"/>	2		NTL team Регистрационный номер:  Персона 36, 	<input checked="" type="checkbox"/>	2021-07-02 05:17:51

<< < Страница 1 > >>

Досье

Досье

Списки наблюдения

Не выбраны

Лица

Все

Силуэты

Все

Автомобили и номера

Все

Регистрационный номер

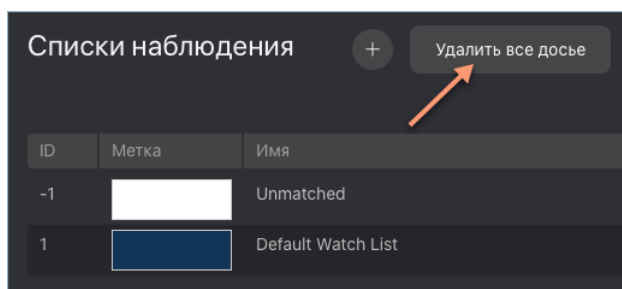
ID досье

Сбросить фильтры

Создать отчет

Очистка базы данных досье

Вы можете очистить базу данных досье в один клик целиком. Для этого перейдите на вкладку *Настройки*. Выберите *Списки наблюдения*. Нажмите *Удалить все досье*.

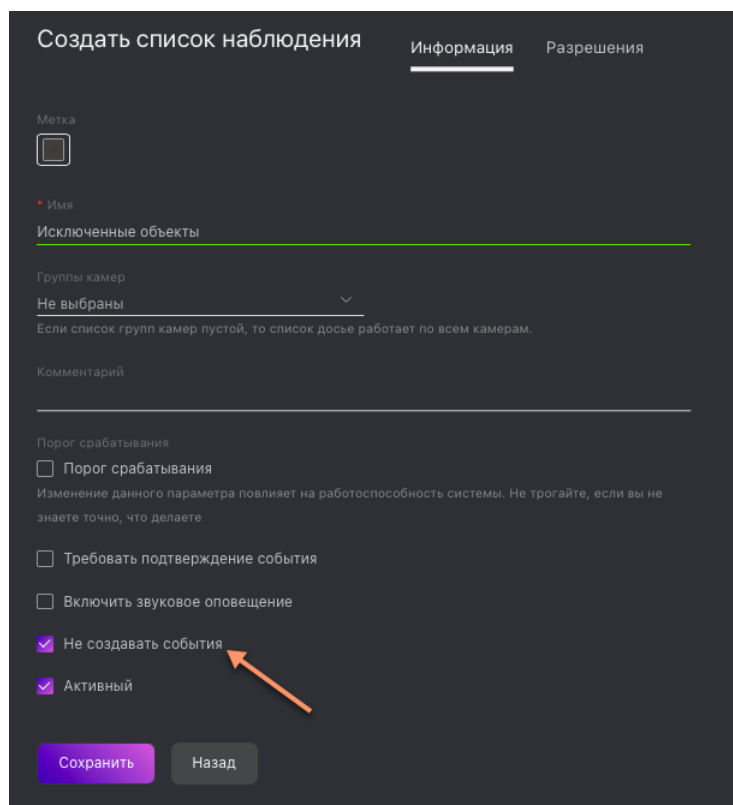


Отключение создания событий для определенных объектов

Иногда при мониторинге необходимо пропускать определенные объекты. Один из самых распространенных случаев — это объекты на рекламных носителях, расположенные в поле зрения камеры. Будучи постоянно обнаруживаемыми 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*.



2. Для каждого такого объекта *создайте досье* и добавьте его в созданный список наблюдения.

2.2.5 Счетчики лиц и силуэтов

Важно: Для подсчета человеческих силуэтов нужно предварительно включить их *детектирование*.

FindFace Multi позволяет подсчитывать лица и силуэты 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 и т. д.

Вы можете считать объекты отдельно по каждой камере или совместно по всем камерам в одной или нескольких группах.

Метод подсчета основан на временных срезах. Это означает, что счетчик считает лица и силуэты на статических скриншотах, которые делаются с заданным интервалом. Счетчик показывает, как количество лиц и силуэтов меняется со временем.

В этом разделе:

- *Настройка счетчиков*
- *Создание счетчика*
- *График счетчика*
- *Работа с записями счетчика*
- *Настройка вебхука для счетчика*

Настройка счетчиков

Для настройки счетчиков откройте файл конфигурации `/etc/findface-security/config.py` и измените следующие параметры:

- `COUNTERS_SAVE_FULLFRAME` определяет параметры сохранения полных кадров при работе счетчиков: `always` (всегда), `detect` - сохранять, только если были обнаружены лица или силуэты, `never` (никогда).
- `COUNTERS_FULLFRAME_JPEG_QUALITY`: качество JPEG полных кадров,
- `COUNTERS_THUMBNAIL_JPEG_QUALITY`: качество JPEG миниатюр.

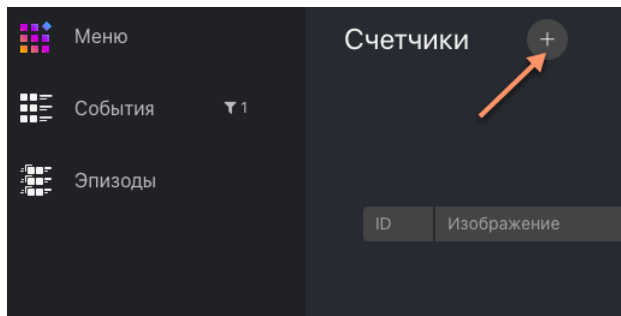
```
sudo vi /etc/findface-security/config.py

# 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,
...
```

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. По умолчанию подсчет будет проводиться по всем камерам в этих группах. При необходимости отключите подсчет по определенным камерам.
5. Задайте интервал 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. Убедитесь, что для счетчика установлен флажок *Активный*.
9. Нажмите *Сохранить*. В результате появятся две новые вкладки.
10. (Опционально) Перейдите на вкладку *Области интереса*, чтобы задать регион отслеживания лиц и силуэтов в поле зрения камер(ы).

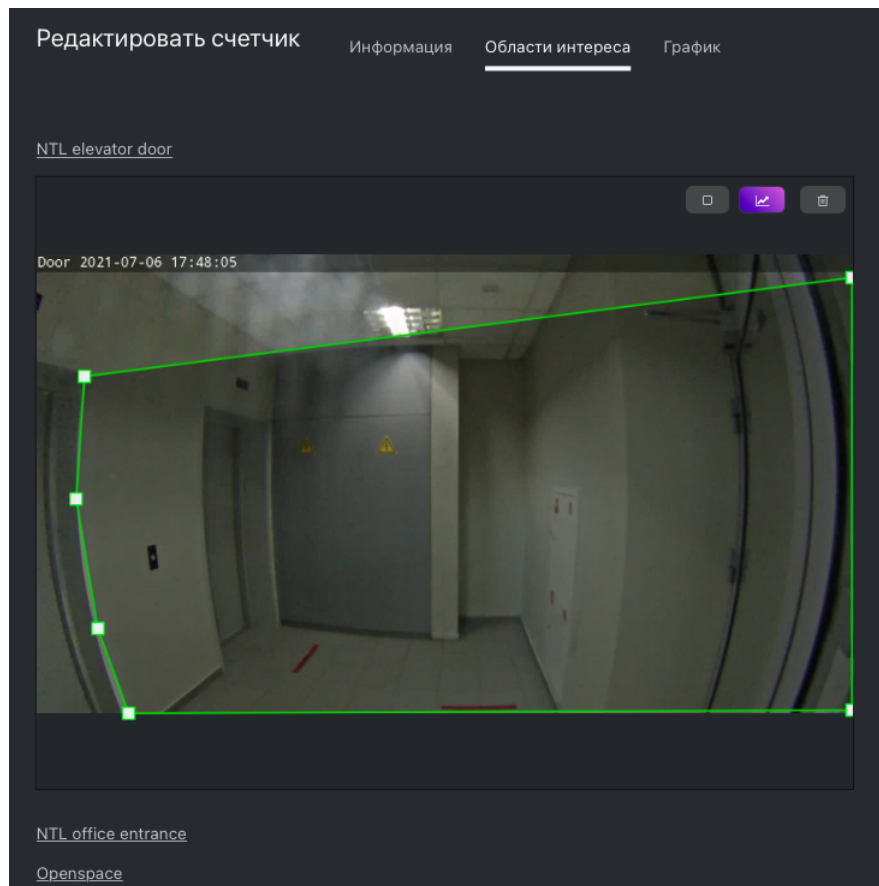
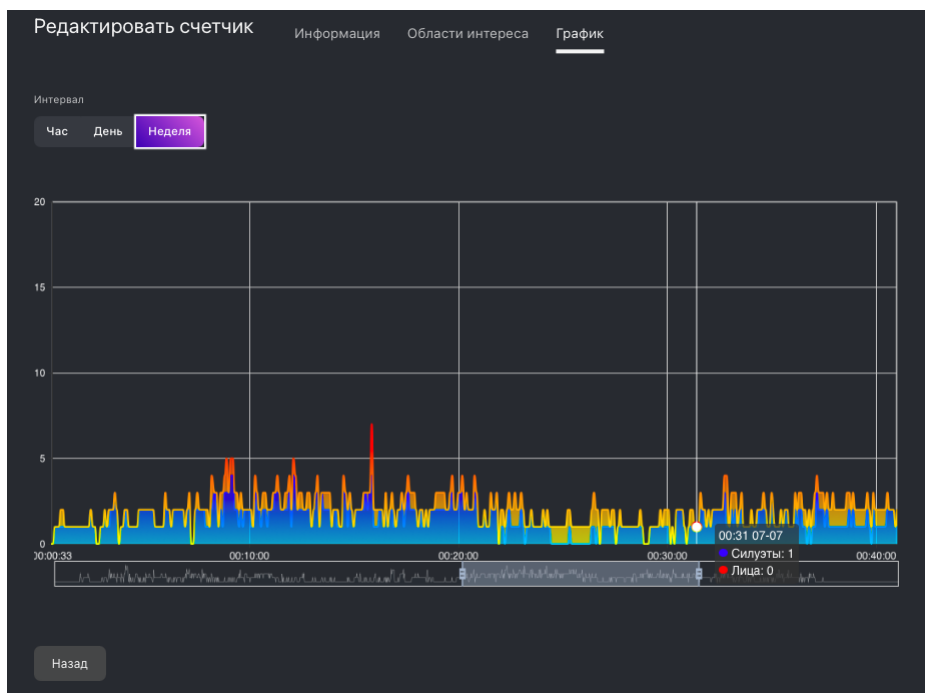


График счетчика

Для того чтобы увидеть график счетчика за последние час, сутки или неделю, перейдите на вкладку *График* в настройках счетчика.



Работа с записями счетчика

Статические скриншоты, сделанные счетчиком, с указанием количества лиц и силуэтов на них сохраняются как т. н. записи счетчика.

Для просмотра записей счетчика перейдите на вкладку *Счетчики*. В соответствующей счетчику строке нажмите на любой столбец, кроме ID (ведет в настройки счетчика).

Записи счетчика				
« < Страница 1 > »				
Id	Изображение	Лица	Силуэты	Дата
396		5	0	2020-08-20 22:09:00
395		2	0	2020-08-20 22:08:55
394		3	0	2020-08-20 22:08:50
393		3	0	2020-08-20 22:08:45

Счетчик: Вход 3

Камеры: Не выбраны

Группы камер: Не выбраны

Старт: 0

Конец: 0

Количество лиц: От

Количество силуэтов: От

Id:

Сбросить фильтры

Для работы с записями счетчика используйте следующие фильтры:

- Счетчик
- Камеры
- Группы камер
- Период времени
- Количество лиц в записи
- Количество силуэтов в записи
- Id записи

Настройка вебхука для счетчика

Для того чтобы автоматизировать процесс сбора статистики, *настройте вебхук* на срабатывание при определенном количестве лиц и силуэтов в записи счетчика.

См.также:

- *Распознавание силуэтов и их атрибутов*
- *Вебхуки*

2.2.6 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.

Важно: В текущей версии функция Эпизоды поддерживает только лица.

Совет: Поиск объекта в списке событий и базе данных досье выполняется на вкладке *Поиск*.

Совет: Для идентификации объектов в архивных видео см. *Идентификация объектов в офлайн видео*.

Работа с событиями

Данный раздел посвящен вкладке *События*.

Совет: *Эпизоды* позволяют поднять безопасность на новый уровень.

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 идентификации в режиме реального времени
- Карточка события. Принятие события
- Карточка события. Поиск объекта


Просмотр событий идентификации в режиме реального времени

Как только объект будет обнаружен, вы увидите уведомление в одном из списков событий: *Лица*, *Силуэты* или *Автомобили*, в зависимости от типа объекта.

Уведомление содержит следующую информацию:

- Если совпадение не найдено: нормализованное изображение объекта, дата и время обнаружения объекта, группа камер.
- Если совпадение найдено: нормализованное изображение объекта, фотография из досье, имя персоны, степень схожести объектов, комментарий из досье, список наблюдения, дата и время обнаружения объекта, группа камер.

Примечание: Система может быть настроена таким образом, что уведомления будут выводиться только для объектов с совпадением с досье.

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.

- *Досье:* отображать только события по определенному досье.
- *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.

Специальные фильтры для лиц

- *ID эпизода:* отобразить события из эпизода с определенным ID.
- *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

- *Цвет одежды, верх:* отображать только события с людьми, одетыми в одежду заданного цвета (верх).
- *Цвет одежды, низ:* отображать только события с людьми, одетыми в одежду заданного цвета (низ).

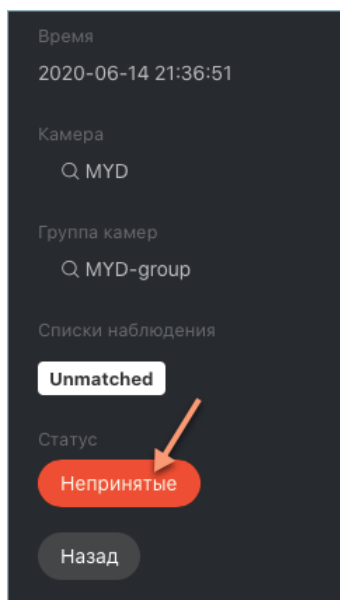
Специальные фильтры для автомобилей

- *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.


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 досье).

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*.



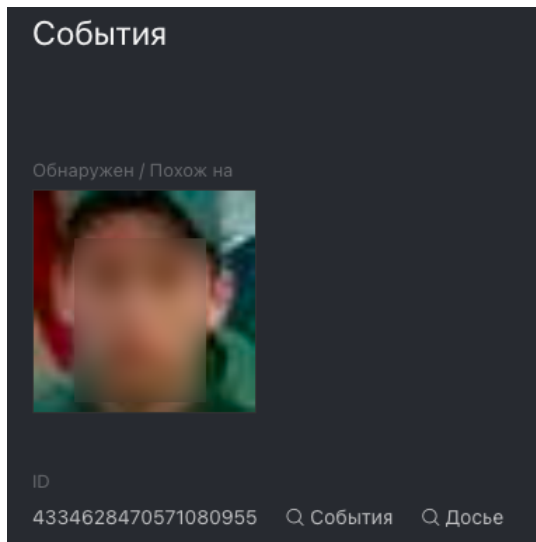
Совет: Если на обнаруженный объект заведено досье, в него можно перейти, щелкнув по имени персоны в карточке события.

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 базе данных досье. Для перехода на вкладку поиска из карточки события нажмите *События* или *Досье*.



См.также:

- [search](#).

Эпизоды событий

Данный раздел посвящен вкладке *Эпизоды*.

См.также:

- [Работа с событиями](#)

Эпизод — это набор событий идентификации, в которых фигурируют лица одного и того же человека, 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 Эпизоды позволяет автоматически объединять входящие события на основе времени обнаружения и схожести лиц. Это позволяет с легкостью обрабатывать разнородные события даже в больших количествах.

В этой главе:

- [Об эпизодах](#)
- [Параметры эпизода](#)
- [Назначение прав на эпизоды](#)
- [Просмотр эпизодов](#)
- [Принятие события и эпизода](#)
- [Фильтрация событий по ID эпизода](#)

Об эпизодах

Эпизод — это набор событий идентификации, в которых фигурируют лица одного и того же человека, обнаруженные в течение определенного периода времени.

Эпизоды бывают двух типов:

- LIVE: открытый на данный момент эпизод, в который могут добавлены новые события.
- Закрытый: закрытый эпизод, добавление событий невозможно.

Параметры эпизода

Для настройки эпизодов вам понадобится файл конфигурации `/etc/findface-security/config.py`. Измените следующие параметры в секции `FFSECURITY`:

- `EPISODE_SEARCH_INTERVAL`: период времени, предшествующий событию, в течение которого система ищет в базе данных векторов признаков события с похожими лицами. Если такого события не найдено, система создает новый эпизод. В противном случае она выбирает наиболее подходящее событие из открытого (LIVE) эпизода, отсортировав 100 последних похожих лиц.

Примечание: Порог срабатывания в эпизодах отличается от порога при верификации лиц в событиях. См. *Основные настройки*.

- `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 лицом. Использование данной опции помогает экономить дисковое пространство.

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

...

FFSECURITY = {
    ...
    'EPISODE_KEEP_ONLY_BEST_EVENT': True,
    'EPISODE_SEARCH_INTERVAL': 60,
    'EPISODE_MAX_DURATION': 300,
    'EPISODE_EVENT_TIMEOUT': 30,
    ...
}

...
```


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: Чтобы увидеть несопоставленные с досье события, вам понадобятся только права на камеру.

Для управления правами на объект **Эпизод** перейдите в разрешения для соответствующей роли и настройте разрешение **eventepisode**.

Совет: См. *Управление пользователями*.

Информация

Списки наблюдения

Группы камер

Разрешения

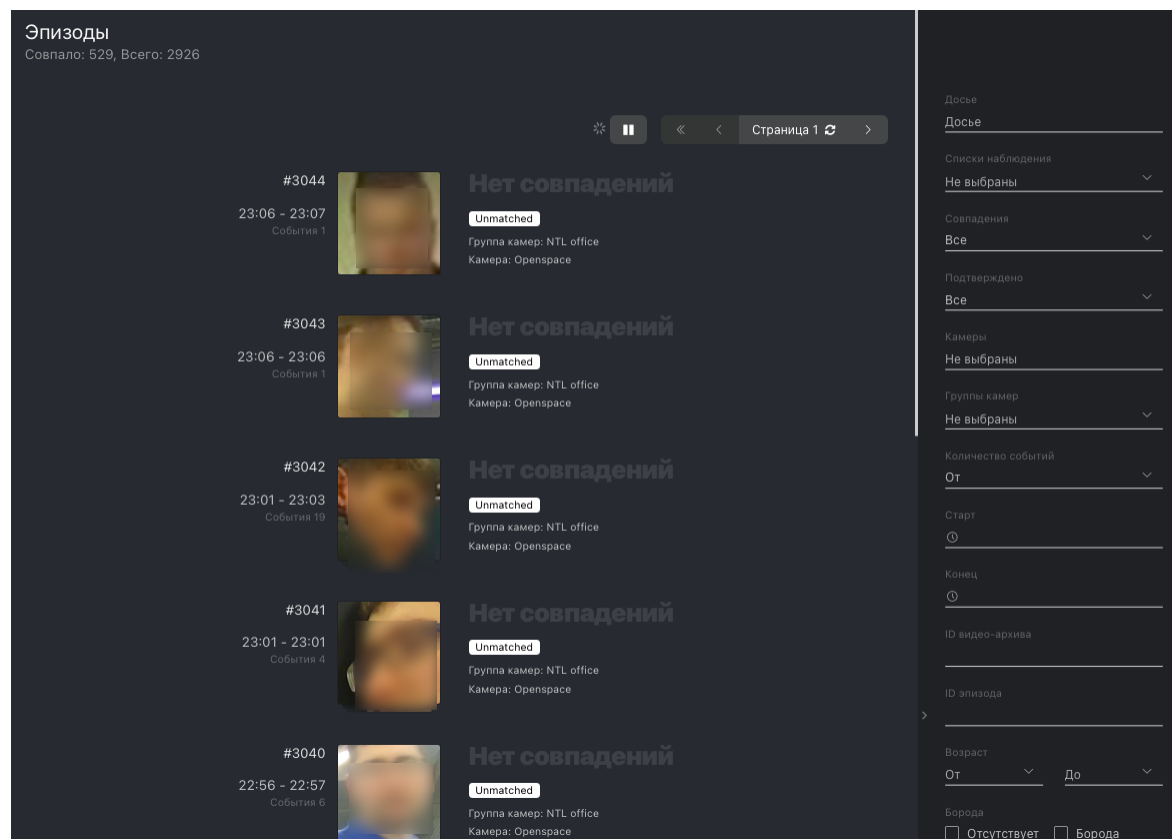
Выбрать все

Отменить все

Имя	Просмотр	Изменить	Добавить	Удалить
faceevent				
faceobject				
carevent				
carobject				
bodyevent				
bodyobject				
dossierlist				
dossier				
cameragroup				
camera				
eventepisode				
person				
uploadlist				
upload				
user				
webhook				
videoarchive				
counter				
metadictionary				
notification				
report				
Имя	Активный			
configure_ntls				
batchupload_dossier				
view_runtimesetting				
change_runtimesetting				
view_auditlog				

Просмотр эпизодов

Список эпизодов с фильтрами и статистикой отображается на вкладке *Эпизоды*. Событие обнаружения лица либо добавляется в существующий LIVE-эпизод, либо инициирует создание нового эпизода. Каждому эпизоду присваивается id, который впоследствии можно использовать для фильтрации событий и эпизодов.

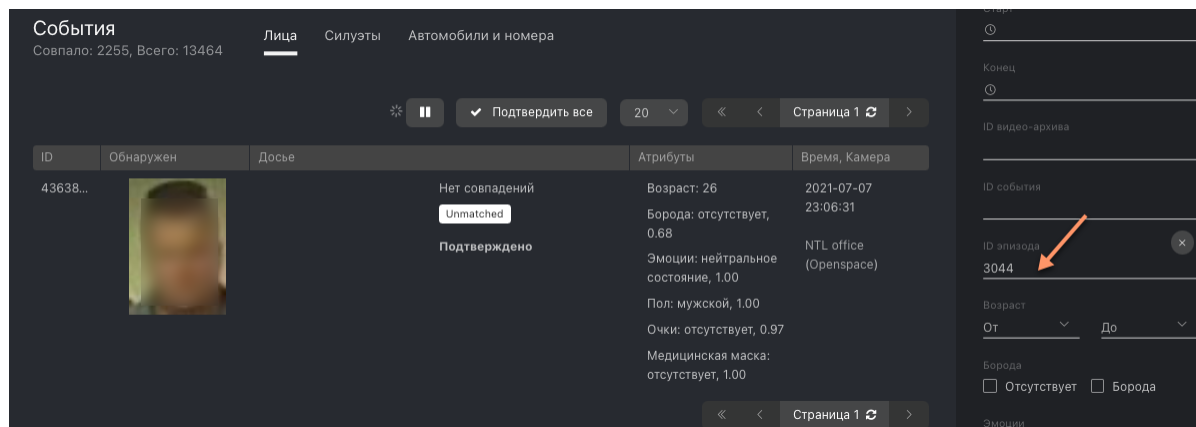


К эпизодам в списке можно применить следующие фильтры:

- *Досье*: отображать только эпизоды по определенному досье.
- *Списки наблюдения*: отображать только эпизоды по определенному списку наблюдения.
- *Совпадения*: отображать только эпизоды с совпадением лиц/без совпадения лиц или все эпизоды.
- *Подтверждено*: отображать только принятые/непринятые или все эпизоды.
- *Камеры*: отображать только эпизоды по определенной камере.
- *Группы камер*: отображать только эпизоды по определенной группе камер.
- *Старт, Конец*: отображать 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.

Вы также можете отфильтровать эпизоды по Liveness и атрибутам лица (если это применимо к вашей системе).

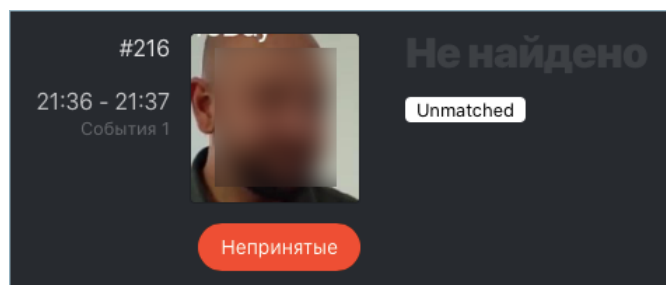
Для просмотра событий щелкните по нужному эпизоду в списке. Вы будете переправлены на вкладку *События* с соответствующим ID эпизода в фильтре *Эпизод*:



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*.

2.2.7 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:

- *Настройка обработки видеофайлов*
- *Обработка видеофайла*

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

...

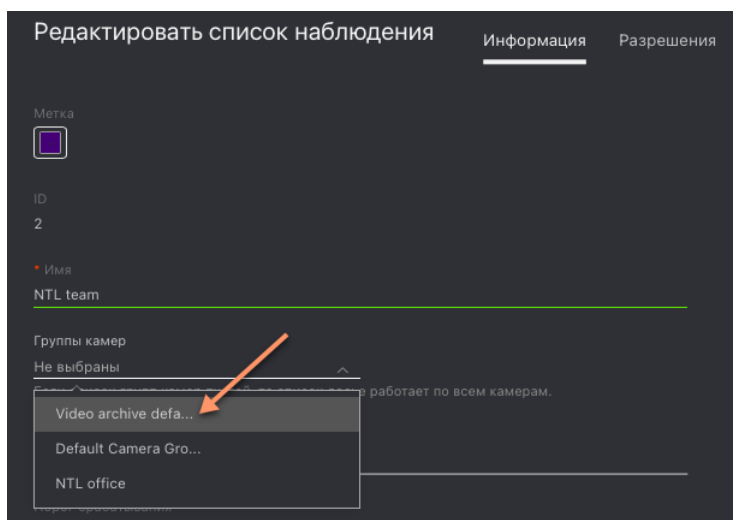
FFSECURITY = {
    ...
    # maximum concurrent video manager jobs for video archives processing
    'MAX_VIDEO_ARCHIVE_JOBS': 3,
    ...
}

...
```

Обработка видеофайла

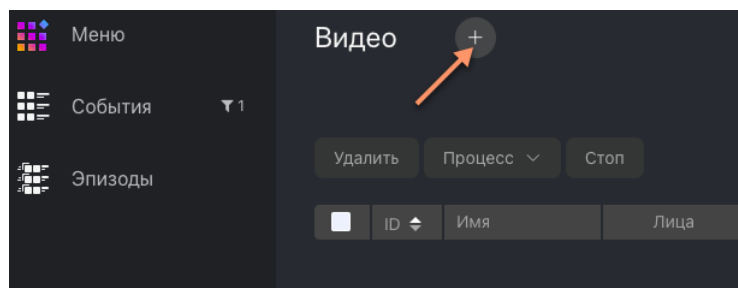
Важно: Перед обработкой видеофайла необходимо выполнить следующие подготовительные действия:

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.

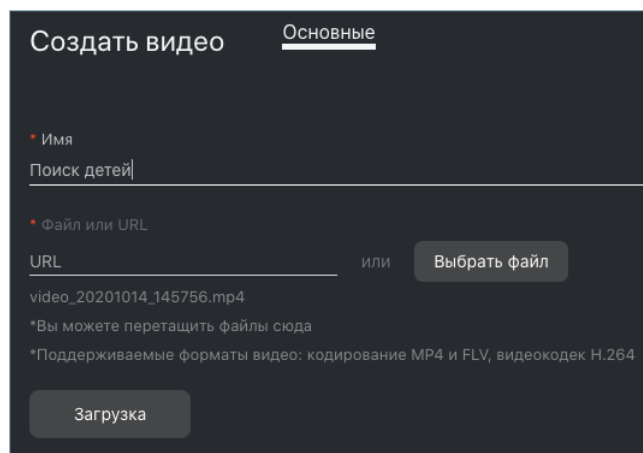


Для идентификации объекта в офлайн видео выполните следующие действия:

1. Создайте видео в FindFace Multi, загрузив его из файла или онлайн-хранилища/облака. Для этого, перейдите на вкладку *Видео*.
2. Нажмите +.

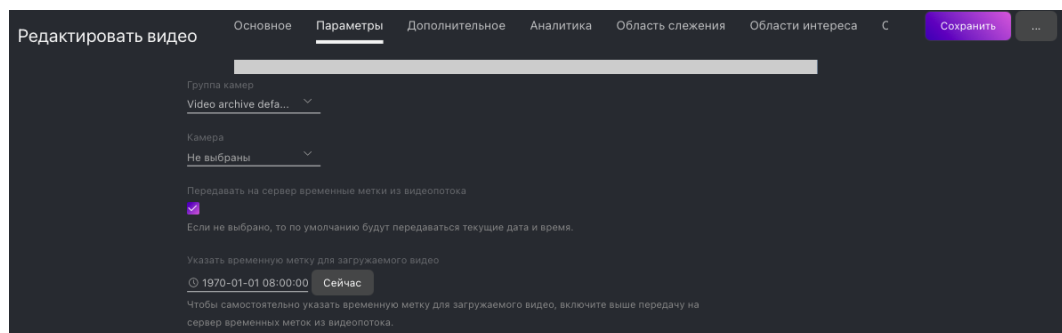


3. Укажите название видео.

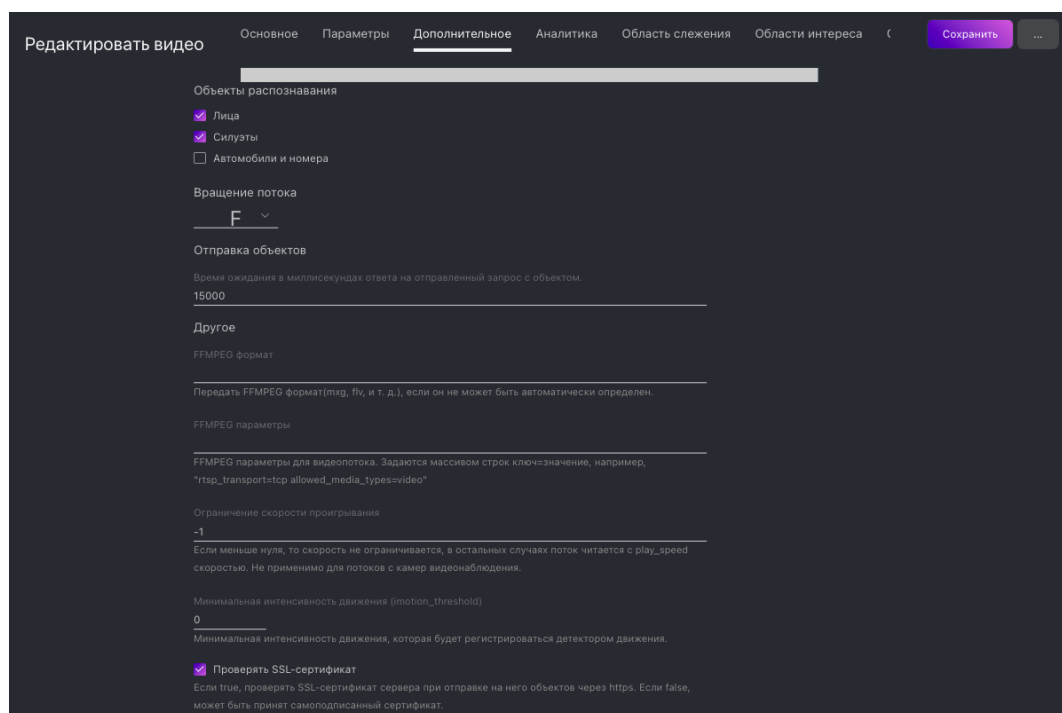


4. Укажите URL видеоизображения в онлайн-хранилище или выберите видеофайл.

5. Нажмите *Загрузка*.
6. По завершении загрузки видео перейдите на вкладку *Параметры*. Задайте параметры обработки видео:



- 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:



- 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.
- *Проверять 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.

Редактировать видео

Основное Параметры Дополнительное **Аналитика** Область слежения

Лица

Минимальное качество изображения объекта (filter_min_quality)
 0.45 0.45
 Минимальное качество изображения объекта для детекции. Допустимый диапазон от 0 до 1. Значения вблизи 1 соответствуют наиболее качественным выровненным изображениям объектов, полностью находящихся в кадре. Значения вблизи 0 соответствуют замыленным изображениям, а также изображениям, содержащим перевернутые объекты или объекты, 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.

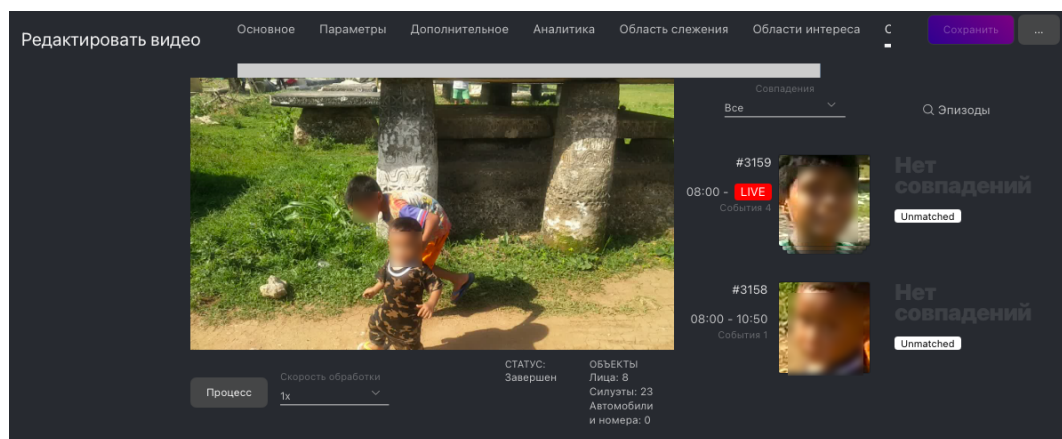
Качество сжатия (jpeg_quality)
 95 95
 Качество сжатия полного кадра для отправки.

Буферный режим (overall_only)
☒ true
 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.

- *Минимальное качество изображения объекта*: Минимальное качество изображения объекта для отправки на сервер. 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 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.
- (Опционально) На вкладках *Область слежения* и *Области интереса* укажите регион отслеживания объектов в поле зрения камеры и зоны детекции для каждого типа объектов.
 - 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.

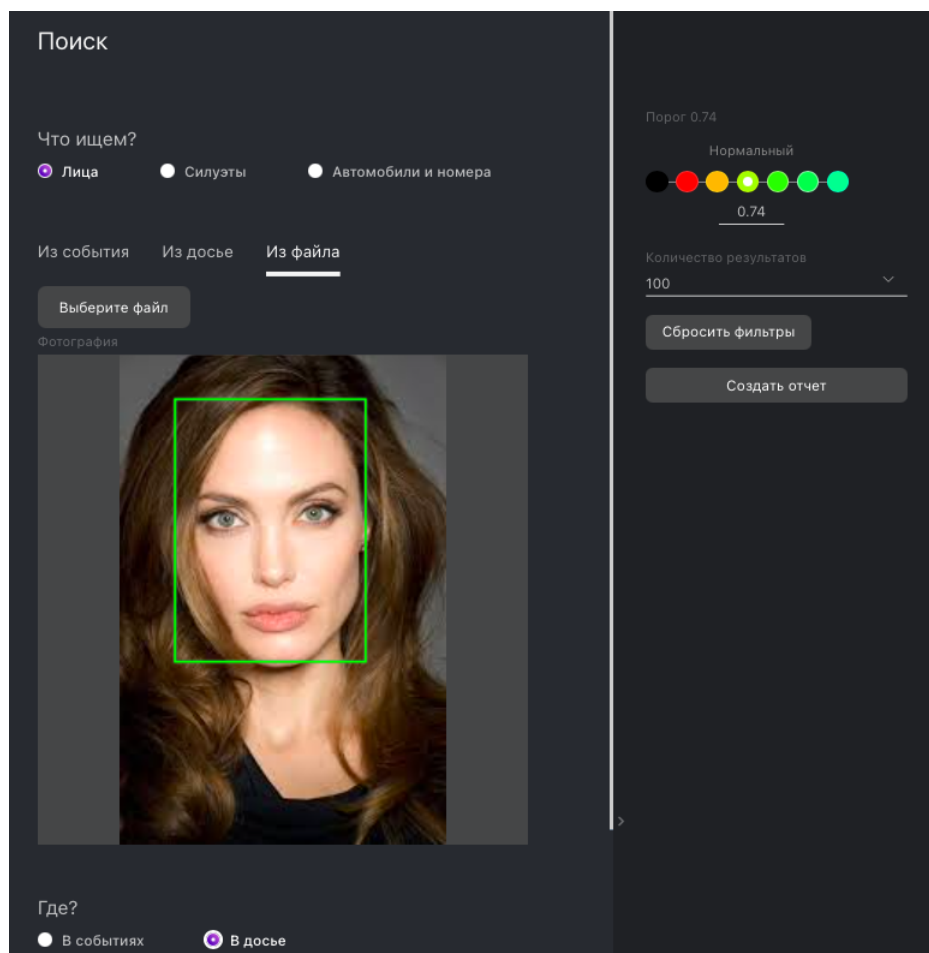
2.2.8 Поиск объектов в базах данных

FindFace Multi позволяет искать объекты в следующих базах данных:

- База данных обнаруженных объектов (вкладка *События*).
- База данных досье (вкладка *Досье*). Содержит эталонные изображения объектов.

Для поиска объекта в базе данных выполните следующие действия:

1. Перейдите на вкладку *Поиск*.

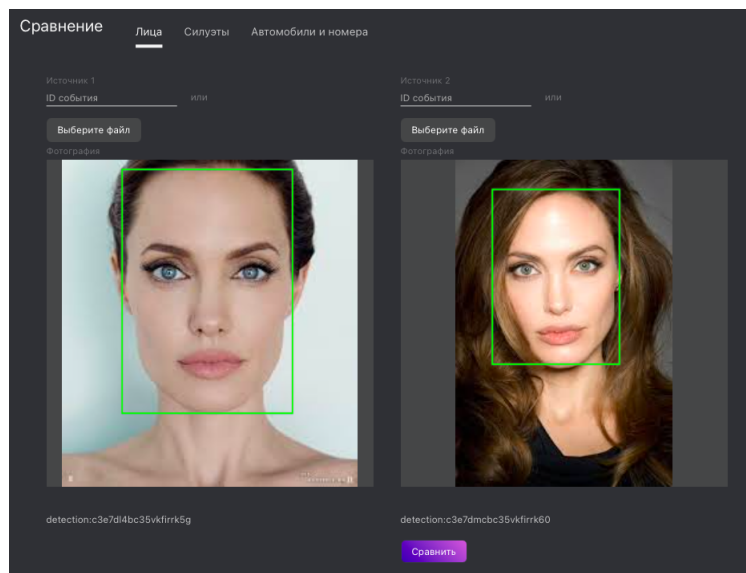


2. Выберите тип искомых объектов: *Лица*, *Автомобили и номера*, *Силуэты*.
3. Укажите базу данных для поиска: *События*, *Досье*.
4. Задайте искомый объект одним из следующих способов:
 - Указав ID события с искомым объектом.
 - Указав ID досье с искомым объектом. Если досье содержит несколько фотографий, выберите те, которые будут использованы для поиска.
 - Загрузив фотографию. Фотография будет отображена в одноименном поле. Если на фотографии присутствует несколько объектов, выберите нужный.
5. По умолчанию система ищет объекты, используя предустановленный порог идентификации, разный для объектов разного типа. При необходимости установите свое собственное значение в фильтре *Порог*.
6. Укажите максимальное количество событий в результатах поиска.
7. Нажмите *Поиск*. Результаты поиска будут отображены ниже. Для каждого найденного объекта будет указана вероятность его совпадения с заданным.

2.2.9 Сравнение двух объектов

FindFace Multi позволяет сравнить два объекта и убедиться в их совпадении. Выполните следующие действия:

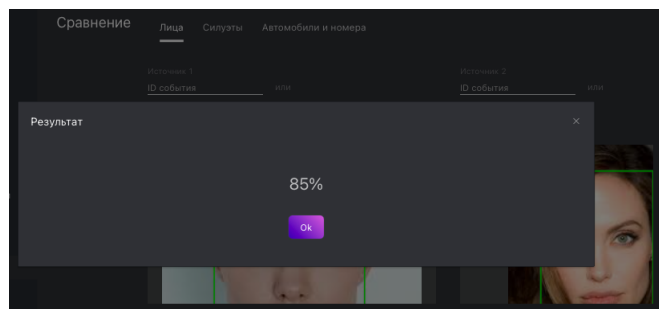
1. Перейдите на вкладку *Настройки*. Нажмите *Сравнение*. Выберите нужную вкладку, в зависимости от типа объектов, которые вы собираетесь сравнить: *Лица*, *Автомобили и номера*, *Силуэты*.



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.



2.2.10 Распознавание персон и связанная с людьми аналитика

FindFace Multi является идеальным инструментом для сбора связанной с людьми аналитики. Включите распознавание персон и используйте его для извлечения значимой информации посредством набора аналитических функций.

Распознавание персон

FindFace Multi может автоматически распознавать персоны. Система на лету распознает лица, принадлежащие одному и тому же человеку, и группирует их с созданием галереи персон. Работа с галереей персон производится на вкладке *Персоны*.

Примечание: Если кластеризация персон включена, в системных базах данных появляется сущность «person event» (событие появления персоны), связанная со всеми *эпизодами*, в которых фигурировало лицо той или иной персоны.

Важно: По умолчанию кластеризация персон отключена. *Включите и настройте ее* через файл конфигурации `/etc/findface-security/config.py`.

В этом разделе:

- *Методы кластеризации*
- *Включение и настройка кластеризации персон*
- *Работа с галереей персон*

Методы кластеризации

FindFace Multi использует следующие методы группировки лиц, принадлежащих одному и тому же человеку:

- Динамическая кластеризация. Происходит на лету после закрытия эпизода. Результат динамической кластеризации отображается в реальном времени на вкладке *Персоны*.

Примечание: Технические детали заключаются в следующем. Не каждый эпизод является подходящим: число событий в нем должно быть равно или больше значения параметра `PERSON_EVENT_MIN_EPISODE_EVENTS` (задается через файл конфигурации `/etc/findface-security/config.py`). Если эпизод соответствует данному требованию, система выбирает событие наилучшего качества и выполняет следующие операции:

- Создает новую сущность `PersonEvent` в основной базе данных PostgreSQL. Сущность содержит метаданные события, вектор признаков лица и миниатюру лица, а также ссылку на родительский эпизод.
- Ищет похожий центроид лица в галерее `person_events` в базе векторов признаков Tarantool. Центроид лица - это виртуальный вектор признаков, усредненный по всем лицам персоны, обнаруженным на данный момент. Если подобный центроид найден, система обновляет его,

используя вектор признаков из нового события. В противном случае создается новый центроид.

- Кластеризация по расписанию. Мы рекомендуем планировать ее на поздние ночные часы, так как процесс занимает много времени и ресурсов.

Примечание: Расписание задается в формате RRULE в параметре PERSONS_CLUSTERIZATION_SCHEDULE файла конфигурации `/etc/findface-security/config.py`. В остальном техническая реализация напоминает динамический метод. Однако качество центроида лица при кластеризации по расписанию лучше, поскольку в этом случае центроиды усредняются по большему массиву накопленных векторов признаков.

Важно: Кластеризация по расписанию полностью перезаписывает содержимое галереи персон, включая id.

Включение и настройка кластеризации персон

По умолчанию кластеризация персон отключена. Для включения данной функции откройте файл конфигурации `/etc/findface-security/config.py` и внесите следующие изменения в раздел SERVICES:

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

...
SERVICES = {
    "ffsecurity": {
        ...
        "persons": True,
    }
    ...
}
```

В результате в веб-интерфейсе FindFace Multi появится вкладка *Персоны*.

В том же файле конфигурации можно изменить следующие параметры:

- PERSON_EVENT_MIN_QUALITY: минимальное качество лиц, используемых в кластеризации персон.
- PERSON_EVENT_MIN_EPISODE_EVENTS: минимальное количество событий в эпизодах, участвующих в кластеризации персон.
- PERSONS_CONFIDENCE_THRESHOLD: пороговая степень схожести для принятия решения о принадлежности лица персоне.

Предупреждение: Перед изменением данного параметра проконсультируйтесь с нашими специалистами по адресу support@ntechlab.com.

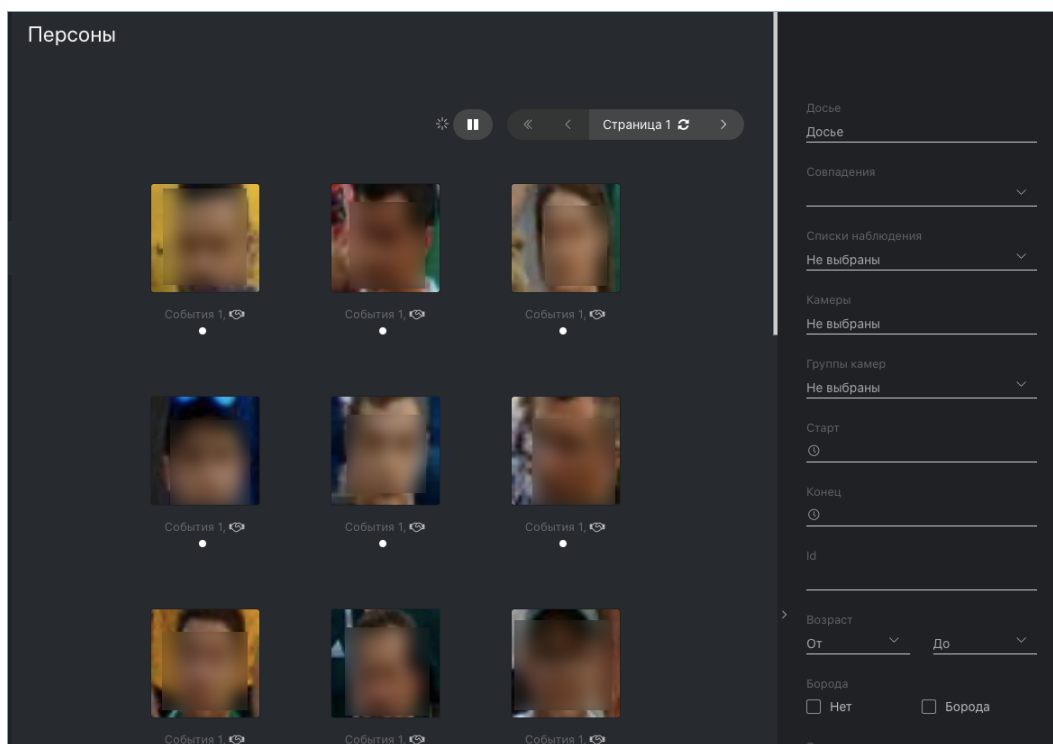
- PERSONS_CLUSTERIZATION_SCHEDULE: рекуррентное правило (RRULE), задающее расписание плановой кластеризации персон.

Совет: См. калькулятор RRULE.

```
# -- Persons configuration --
# rrule (recurrence rule) for scheduling persons clusterization
# WARNING: all scheduling works with UTC time and NOT aware of any timezone
'PERSONS_CLUSTERIZATION_SCHEDULE': 'RRULE:FREQ=DAILY;INTERVAL=1;WKST=MO;BYHOUR=0;
↳BYMINUTE=0',
# face to person matching confidence threshold
'PERSONS_CONFIDENCE_THRESHOLD': 0.739,
# minimum required face quality for person creation
'PERSON_EVENT_MIN_QUALITY': 0.45,
# minimum required number events in episode for person creation
'PERSON_EVENT_MIN_EPISODE_EVENTS': 1,
```

Работа с галерей персон

Для просмотра галереи персон перейдите на вкладку *Персоны*.



Для работы с галереей персон используйте следующие фильтры:

- Досье
- Совпадения
- Камеры
- Группы камер
- Списки наблюдения
- Период времени
- Id персоны

- Атрибуты лица (если используются)
- Liveness (если используется)

См.также:

- *Файл конфигурации findface-security*
- *Вебхуки*
- *Анализ социальных взаимодействий*
- *Видеоаналитика*

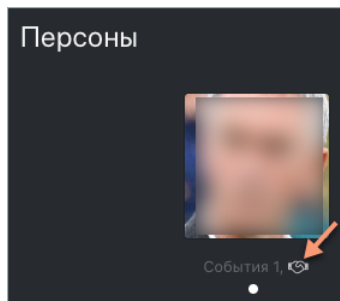
Анализ социальных взаимодействий

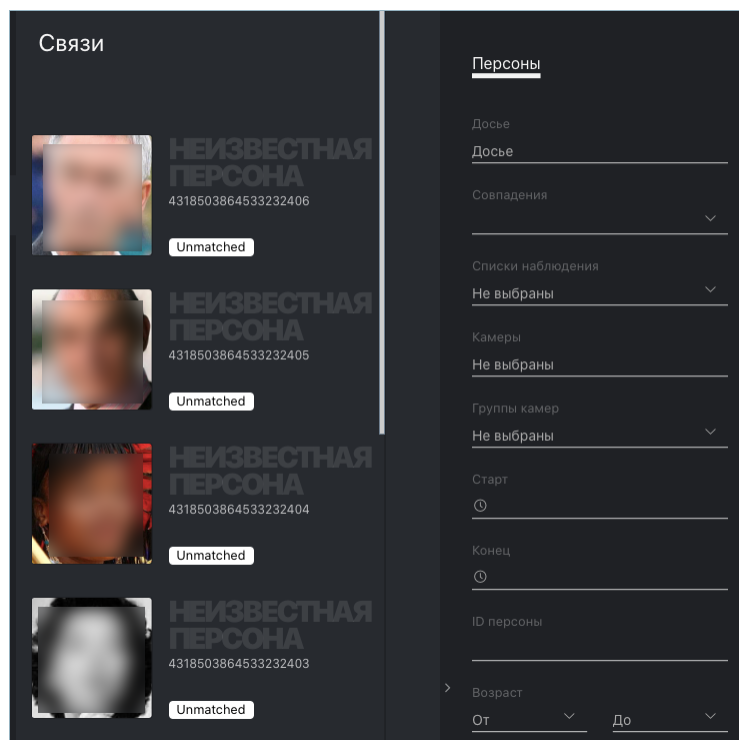
Вы можете посмотреть круг людей, с которыми человек до этого контактировал. Для каждого человека из первого круга система ищет еще один круг связанных с ним людей и т. д. В целом, социальное взаимодействие анализируется до трех кругов в глубину.

Важно: Анализ социальных взаимодействий работает только при включенном *распознавании персон*.

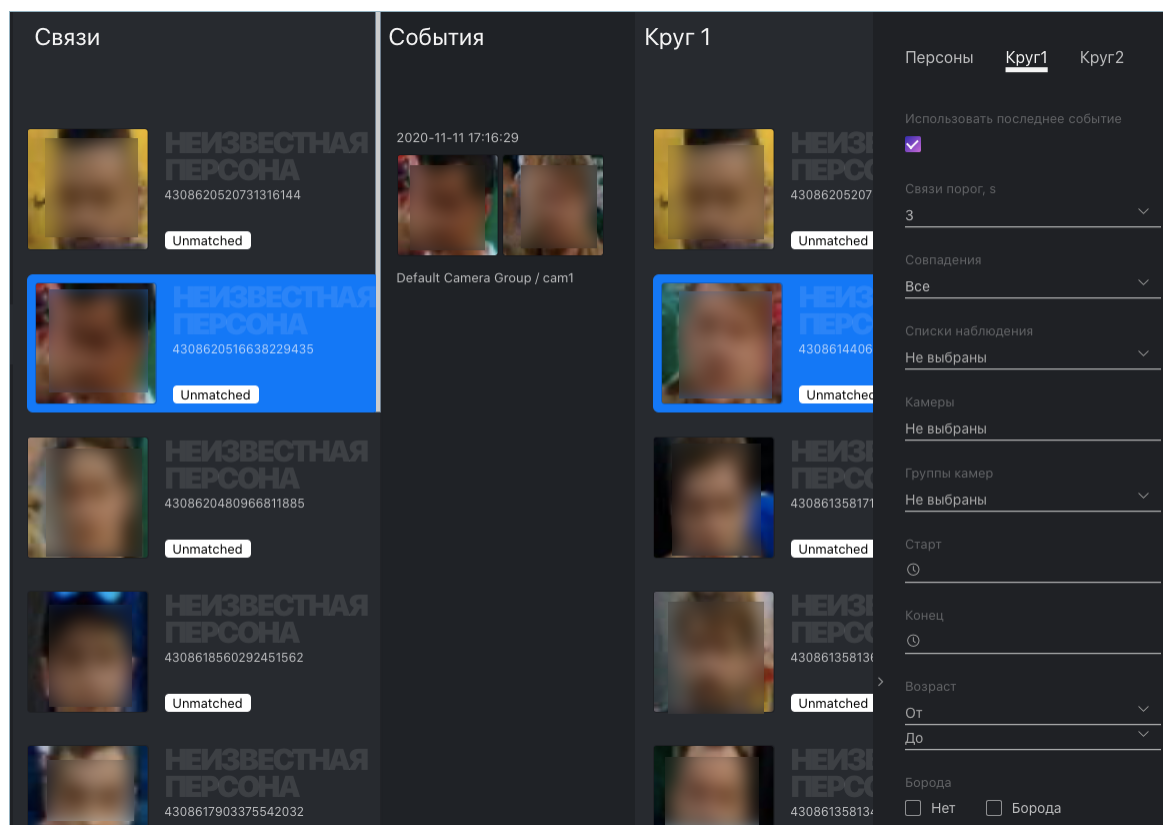
Анализ социальных взаимодействий доступен на вкладке *Связи*.

Совет: Вы также можете отобразить круг связанных людей непосредственно из вкладки *Персоны*, нажав на значок рукопожатия.





На вкладке *Связи* нажмите на изображение персоны, чтобы отобразить первый круг связей. Продолжайте, пока не раскроете все дерево социальных взаимодействий.

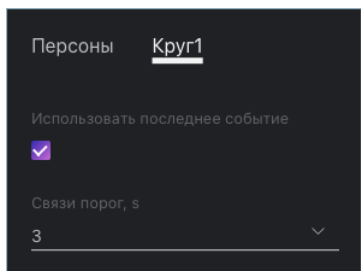


Фильтры можно применять к каждому кругу.

Совет: Например, вы можете найти всех пожилых людей или людей без маски, кто прямо или косвенно взаимодействовал с потенциально инфицированным человеком.

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.

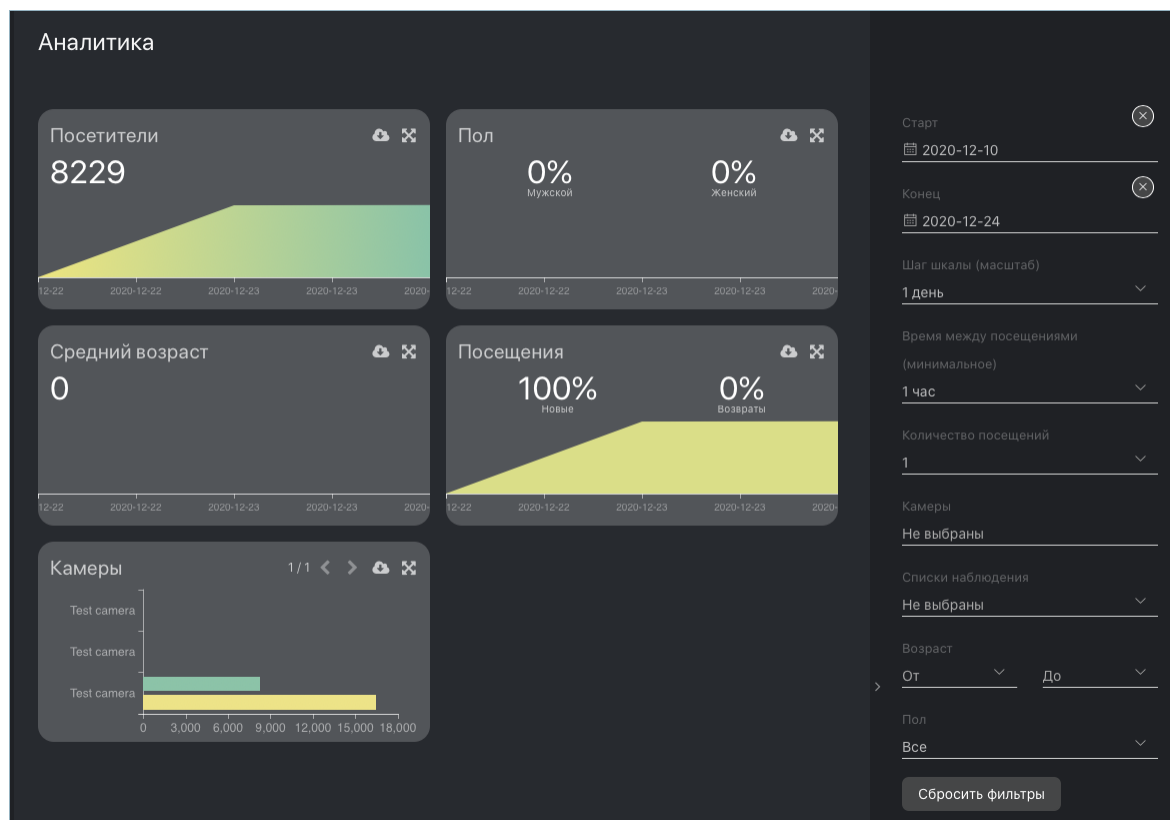


Видеоаналитика

Видеоаналитика 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 «Знай своего клиента».

Диаграммы с аналитическими данными доступны на вкладке *Аналитика*.

Важно: Построение аналитики выполняется только при включенном *распознавании персон*.



Для работы с данными аналитики используйте следующие фильтры:

- Период времени
- Шаг шкалы
- Время между посещениями
- Количество посещений
- Камеры
- Списки наблюдения
- Возраст
- Пол

См. также:

- *Распознавание персон*

2.2.11 Отчеты

FindFace Multi позволяет строить отчеты по следующим системным объектам:

- события распознавания объектов
- эпизоды
- события поиска
- персоны

- камеры
- досье
- аналитические данные
- журнал действий пользователей

В этой главе:

- *Настройка сохранения изображений в отчетах*
- *Построение отчета*

Настройка сохранения изображений в отчетах

При создании отчета 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.

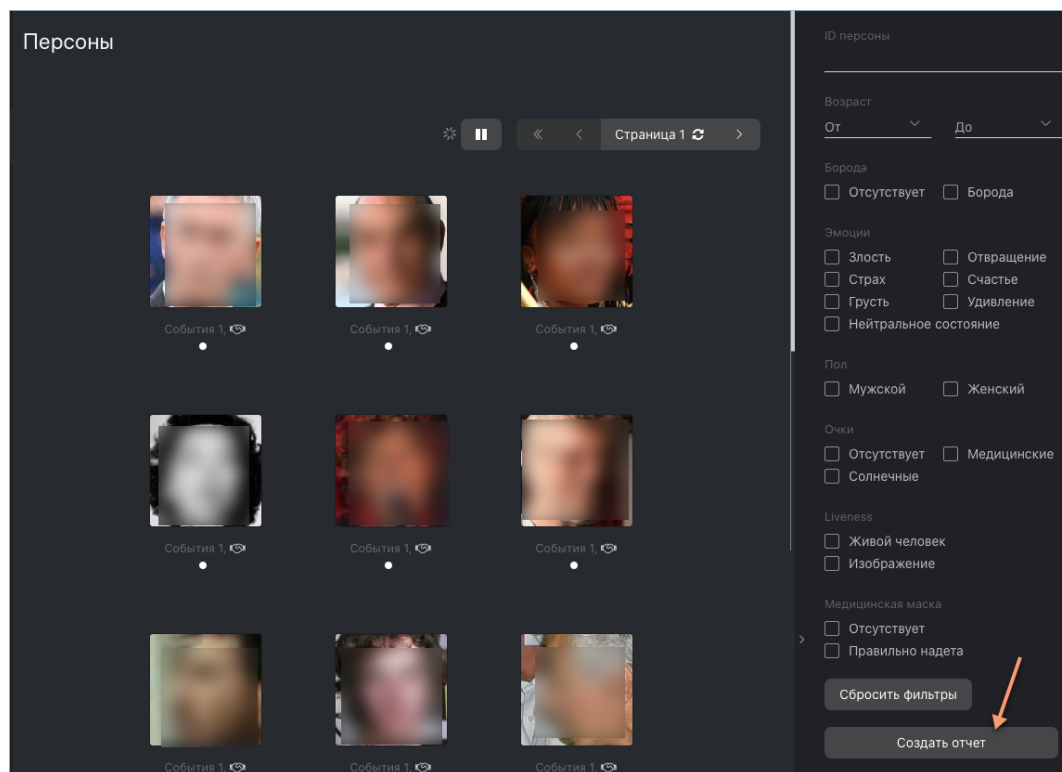
```
sudo vi /etc/findface-security/config.py

# reports image saving options
'REPORT_THUMBNAIL_JPEG_QUALITY': 75,
'REPORT_THUMBNAIL_MAX_HEIGHT': 100,
'REPORT_FULLFRAME_JPEG_QUALITY': 75,
'REPORT_FULLFRAME_MAX_HEIGHT': 250,
```

Построение отчета

Для того чтобы построить отчет, выполните следующие действия:

1. Перейдите на вкладку, соответствующую нужному системному объекту: *События*, *Эпизоды*, *Поиск*, *Персоны*, *Камеры*, *Досье*, *Аналитика*, *Журнал действий*.
2. Установите фильтры для отчета.
3. Нажмите *Создать отчет*.



4. Укажите название отчета. Выберите, следует ли сохранять изображения отчета в виде ссылок, миниатюр или полных кадров. Нажмите *Создать*.

5. Отчет будет доступен для загрузки на вкладке *Отчеты*.

Отчеты

Скачать Обновить Удалить

« < Страница 1 ↻ > »

<input type="checkbox"/>	ID ↕	Имя	Тип	Обновлено ↕	Записи	Размер ↕	Статус
<input type="checkbox"/>	34	Отчет по персонам	Персоны	2020-12-24 13:21:35	9000		Обработка
<input type="checkbox"/>	33	Отчет по персоне 4318271820886733514	Персоны	2020-12-23 19:24:53	2	10.04KB	Завершено Скачать
<input type="checkbox"/>	32	Отчет по персонам	Персоны	2020-12-23 19:21:28	2	6.18KB	Завершено Скачать
<input type="checkbox"/>	30	Отчет по событиям	События	2020-12-22 23:56:29	107	16.82KB	Завершено Скачать
<input type="checkbox"/>	29	Отчет по событиям	События	2020-12-22 23:55:53	107	16.82KB	Завершено Скачать
<input type="checkbox"/>	28	Отчет по поиску	События	2020-12-22 23:54:58	10	7.05KB	Завершено Скачать
<input type="checkbox"/>	26	Отчет по поиску	События	2020-12-22 23:53:02	31	9.33KB	Завершено Скачать
<input type="checkbox"/>	23	Отчет по событиям	События	2020-12-22 23:50:12	0	5.48KB	Завершено Скачать
<input type="checkbox"/>	22	Отчет по событиям	События	2020-12-22 23:46:56	68	12.91KB	Завершено Скачать
<input type="checkbox"/>	21	Отчет по персонам	Персоны	2020-12-22 23:45:56	9	22.25KB	Завершено Скачать

2.2.12 Видеостена

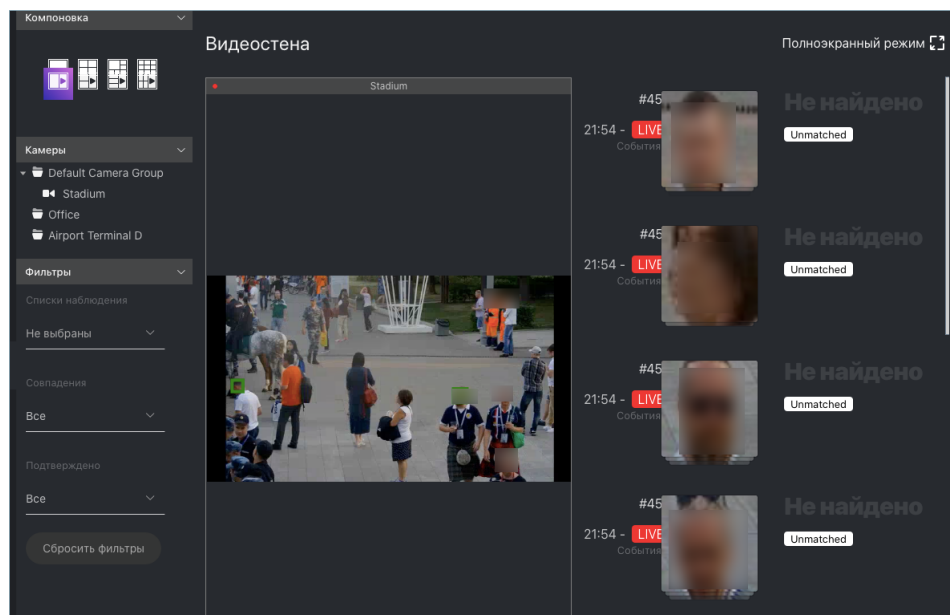
В FindFace Multi встроен базовый функционал видеонаблюдения. Используйте видеостену для отображения видео с камер и видеофайлов.

Видеостена может работать в двух режимах (по 4 раскладки в каждом):

- видеотрансляция,
- видеотрансляция с детектированием лиц и лентой эпизодов.

Для отображения на видеостене видеоизображения выполните следующие действия:

1. Перейдите на вкладку *Видеостена*.
2. Выберите режим работы видеостены 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*.

2.3 Расширенный функционал

2.3.1 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.

- 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
```

- 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.

2.3.2 Распределенная база данных досье

В распределенной архитектуре часто необходимо, чтобы база данных досье распределялась по нескольким серверам.

В текущей реализации база данных досье доступна для редактирования только на главном сервере, т. н. master-сервере. Master-сервер синхронизирован с несколькими дополнительными slave-экземплярами FindFace Multi. На slave-серверах база данных досье доступна только для чтения и мониторинга.

Важно: Досье можно удалить на slave-сервере, если master-сервер недоступен.

Важно: Если список наблюдения на будущем slave-сервере уже содержит досье, синхронизация будет отменена. Убедитесь, что список наблюдения пуст.

Предупреждение: На серверах master и slave должна использоваться одна и та же нейронная сеть.

В этом разделе:

- *Настройка синхронизации Master/Slave*
- *Репликация списка наблюдения master -> slave*
- *Задание времени синхронизации*
- *Отмена репликации и синхронизации списка наблюдения*
- *Включение функционала в веб-интерейсе*

Настройка синхронизации Master/Slave

Для настройки синхронизации master/slave выполните следующие действия:

1. На master-сервере откройте файл конфигурации `/etc/findface-security/config.py`. Придумайте токен синхронизации и укажите его в параметре `SYNC_TOKEN` (предварительно раскомментируйте его).

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

...

# ===== DossierLists sync =====
...
# token must be identical on master and slave
# use pwgen -s 64 1
SYNC_TOKEN = 'ABC_123456789'
...
```

2. На slave-серверах раскомментируйте параметр `SYNC_TOKEN` в файле конфигурации `/etc/findface-security/config.py` и вставьте в него созданный токен синхронизации. Токены на master- и slave-серверах должны быть идентичны.

Синхронизация master/slave теперь настроена и будет активирована, как только вы настроите репликацию списка наблюдения с master-сервера на slave.

Репликация списка наблюдения master -> slave

Для того чтобы реплицировать список наблюдения от master-сервера на slave-экземпляры, отправьте на slave POST-запрос со следующими параметрами в теле:

- `remote_dossier_list`: id исходного списка наблюдения на master-сервере
- `remote_url`: URL master-сервера
- `slave_dossier_list`: id списка наблюдения на slave-сервере, который будет репликой исходного списка наблюдения

```
POST /sync/dossier-lists/
{remote_dossier_list: 1,
remote_url: "http://172.17.46.14",
slave_dossier_list: 3}
```

Задание времени синхронизации

Для того чтобы задать расписание синхронизации досье, выполните следующие действия:

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

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

2. Раскомментируйте и задайте следующие параметры:

- `SYNC_SCHEDULE`: рекуррентное правило (RRULE), задающее расписание синхронизации.

Совет: См. калькулятор RRULE.

- `SYNC_AT_STARTUP`: если `True`, синхронизация выполняется при запуске и перезагрузке FindFace Multi.
- `SYNC_AT_CREATION`: если `True`, синхронизация выполняется сразу после того, как вы ее создали для списка наблюдения.

```
...
# ===== DossierLists sync =====
...

# rrule that defines sync schedule
SYNC_SCHEDULE = 'RRULE:FREQ=DAILY;WKST=MO;BYHOUR=11;BYMINUTE=0'
# if True synchronization will occur on FindFace Security startup and restart
SYNC_AT_STARTUP = True
# if True synchronization will occur immediately after creating synchronization for
↪ dossier list
SYNC_AT_CREATION = True
```

3. Раскомментируйте приведенные выше параметры на каждом slave-сервере. Значения параметров могут быть любыми.

Отмена репликации и синхронизации списка наблюдения

Для того чтобы отменить репликацию и синхронизацию списка наблюдения, отправьте на slave-сервер API-запрос с `{id}` соответствующего списка на slave-сервере:

```
DELETE /sync/dossier-lists/{id}/
```

Включение функционала в веб-интерейсе

По умолчанию вы можете включать и отключать репликацию списка наблюдения только через API. Для того чтобы данный функционал был также доступен через веб-интерфейс, выполните следующие действия:

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

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


- Включите плагин `ffsecurity_sync`, раскомментировав строку `INSTALLED_APPS.append('ffsecurity_sync')` в разделе плагинов:

```
...

# ===== DossierLists sync =====
INSTALLED_APPS.append('ffsecurity_sync')

...
```

- Выполните аналогичные действия на каждом slave-сервере.
- На каждом сервере выполните перенос основной архитектуры базы данных из FindFace Multi в PostgreSQL. Заново создайте группы пользователей в основной базе данных. Перезапустите сервис `findface-security`.

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

2.3.3 Пользовательские вкладки, поля и фильтры в досье

Часто необходимо, чтобы досье в веб-интерфейсе содержало дополнительные вкладки и поля, по которым можно осуществлять поисковые запросы.

См.также:

Для того чтобы создать пользовательские поля досье в базе данных векторов признаков Tarantool, см. *Пользовательские метаданные лиц досье в Tarantool*.

Для добавления пользовательских вкладок и полей в досье выполните следующие действия:

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

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

- В разделе `FFSECURITY` раскомментируйте секцию `CUSTOM_FIELDS` и измените ее примерное содержимое с учетом следующих правил:
 - 'items': список полей в досье. Опишите каждое поле следующими параметрами:
 - 'name': внутреннее имя поля, string.
 - 'label': название поля в веб-интерфейсе, string.
 - 'display': формат отображения (`form` или `list`), string или array.
 - 'tab': вкладка, на которой располагается поле. Если не задана, поле появится на главной странице досье (той, что с фотографией).
 - '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 досье. Первая вкладка в списке соответствует главной странице досье.

```
FFSECURITY = {
...

# Edit CUSTOM_FIELDS section to customize dossier content.
# Below is an example for integration FindFace Security with Sigur.
'CUSTOM_FIELDS': {
    'dossier_meta': {
        'items': [
            {
                'name': 'personid',
                'default': '',
                'label': 'PersonID',
                'display': ['list', 'form'],
                'description': 'Sigur person ID'
            },
            {
                'name': 'firstname',
                'default': '',
                'label': 'First Name',
                'display': ['list', 'form'],
                'description': 'Sigur first name'
            },
            {
                'name': 'lastname',
```

(continues on next page)

(продолжение с предыдущей страницы)

```

        'default': '',
        'label': 'Last Name',
        'display': ['list', 'form'],
        'description': 'Sigur last name'
    },
    {
        'name': 'version',
        'default': '',
        'label': 'Version',
        'display': ['list', 'form'],
        'description': 'Sigur photo version'
    }
],
'filters': [
    {
        'name': 'personid',
        'label': 'Sigur person ID filter',
        'field': 'personid'
    }
]
},

```

4. Перезапустите сервис `findface-security`.

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

5. Вы увидите, что новая форма досье появилась в веб-интерфейсе.

2.3.4 Пользовательские метаданные лиц досье в Tarantool

Часто бывает необходимо задать в базе данных Tarantool дополнительные метаданные для лиц досье.

См.также:

Для того чтобы создать пользовательские вкладки, поля и фильтры в веб-интерфейсе, см. [Пользовательские вкладки, поля и фильтры в досье](#).

Для задания пользовательских метаданных лиц выполните следующие действия:

1. Подготовьте список пользовательских мета-полей, которые будут присвоены лицам досье в Tarantool.
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_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  
            }  
        ]  
    }  
},  
},
```

4. *Добавьте новые мета-поля* в структуру базы данных Tarantool.
5. Вы можете работать с новыми мета-полями через *HTTP API*, используя методы `objects/faces/`.

2.3.5 Пакетная загрузка фотографий через консоль

Помимо веб-интерфейса, для пакетной загрузки фотографий в базу данных досье можно использовать поставляемую вместе с FindFace Multi утилиту `findface-security-uploader`. Используйте утилиту, когда требуется загрузить большое количество фотографий (более 10000).

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
```

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 --dossier-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.
- `-dossier-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 досье как неактивные.
- **--noface:** включить изображения без детектирования.

2.3.6 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 досье, 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.

2.3.7 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 видеофрагмента и возвращает изображение лица наилучшего качества и десятичный результат `liveness`, усредненный 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`.

FindFace Multi использует `findface-liveness-api` для *аутентификации* по лицу. При необходимости вы можете установить и использовать данный сервис отдельно.

В этом разделе:

- *Установка и настройка `findface-liveness-api`*
- *Использование Liveness API*

Установка и настройка `findface-liveness-api`

Сервис `findface-liveness-api` автоматически устанавливается вместе с FindFace Multi.

Для того чтобы установить сервис отдельно, 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
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
```

(continues on next page)

(продолжение с предыдущей страницы)

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


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

2.3.8 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. Остановите исходный сервис `findface-extraction-api-gpu`.

```
sudo service findface-extraction-api stop
```

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 соответствующие номера устройств GPU к именам новых файлов конфигурации, как показано в примере ниже (устройства GPU №0 и №6).

```
/etc/findface-extraction-api@0.ini  
/etc/findface-extraction-api@6.ini
```

3. Откройте новые файлы конфигурации. Укажите номера устройств GPU, а также номера слушающих портов.

```
sudo vi /etc/findface-extraction-api@0.ini  
  
listen: 127.0.0.1:18666  
...  
  
gpu_device: 0  
...
```

```
sudo vi /etc/findface-extraction-api@6.ini  
  
listen: 127.0.0.1:18667  
...  
  
gpu_device: 6  
...
```

4. Запустите новые сервисы.

```
sudo service findface-extraction-api@0 start  
sudo service findface-extraction-api@6 start
```

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
```

2.3.9 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 силуэтов в фотографиях досье.
- `ffsec_car_events`: векторы признаков, извлеченные из обнаруженных на видео автомобилей.
- `ffsec_car_objects`: векторы признаков, извлеченные из автомобилей в фотографиях досье.
- `ffsec_face_events`: векторы признаков, извлеченные из обнаруженных на видео лиц.
- `ffsec_face_objects`: векторы признаков, извлеченные из лиц в фотографиях досье.
- `ffsec_user_face`: векторы признаков, извлеченные из фотографий пользователей FindFace Multi, используемые для аутентификации по лицу.
- `ffsec_persons`: центроиды персон (виртуальные векторы признаков, усредненные по всем лицам персоны) и метаданные.

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 (стопроцентное 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 описанием 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": {
```

(continues on next page)

(продолжение с предыдущей страницы)

```

        "data": "qgI3vZRv/zOBQTk9rcirOyZrNp09MdHavW1WuT0=",
        "score": [0.75, 1]
    },
    "id": {
        "range": [9223372036854000000, 9223372036854999000]
    },
    "meta": {
        "person_id": {
            "range": [444, 999]
        },
        "cam_id": {
            "set": ["12767", "8632", "23989"]
        }
    }
}
}'

```

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/zOBQTk9rcirOyZrNp09MdHavW1WuT0=",
      "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)
```

(continues on next page)

(продолжение с предыдущей страницы)

```
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/1.1 409 Conflict
Content-length: 57
Server: Tarantool http (tarantool v1.7.3-673-g23cc4dc)
Connection: keep-alive

{"error":{"message":"gallery already exists","code":409}}
```

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
```

2.3.10 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 лиц, для которых совпадения не были найдены. Для этого установите `'BLUR_UNMATCHED_FACES': True`. При желании вы можете изменить установленное по умолчанию для данных кадров качество JPEG.

```
...

FFSECURITY = {
    ...

    # blur all unmatched faces on the full frame of the matched event (GDPR support)
    'BLUR_UNMATCHED_FACES': False,

    # full frame jpeg quality when `BLUR_UNMATCHED_FACES` is enabled
```

(continues on next page)

(продолжение с предыдущей страницы)

```
'BLURRED_FULLFRAME_JPEG_QUALITY': 85,  
...  
}
```

4. Перезапустите `findface-security`.

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

2.3.11 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*.

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.

3.1 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*.

3.2 Вебхуки

Вы можете настроить FindFace Multi для автоматической отправки уведомлений об определенных событиях, эпизодах, записях счетчика на заданный URL-адрес. Для этого создайте и настройте вебхук. При наступлении нужного события/эпизода/записи счетчика FindFace Multi отправит HTTP-запрос на URL-адрес, указанный в настройках вебхука.

Вебхуки можно использовать для решения разнообразных задач, например, для уведомления пользователя об определенном событии, вызова определенных действий на целевом веб-сайте, при решении задач безопасности, таких как удаленное автоматическое управление доступом и др.

В этом разделе:

- *Настройка вебхука*
- *Как работает вебхук*
- *Подробные данные в вебхуках (Verbose)*

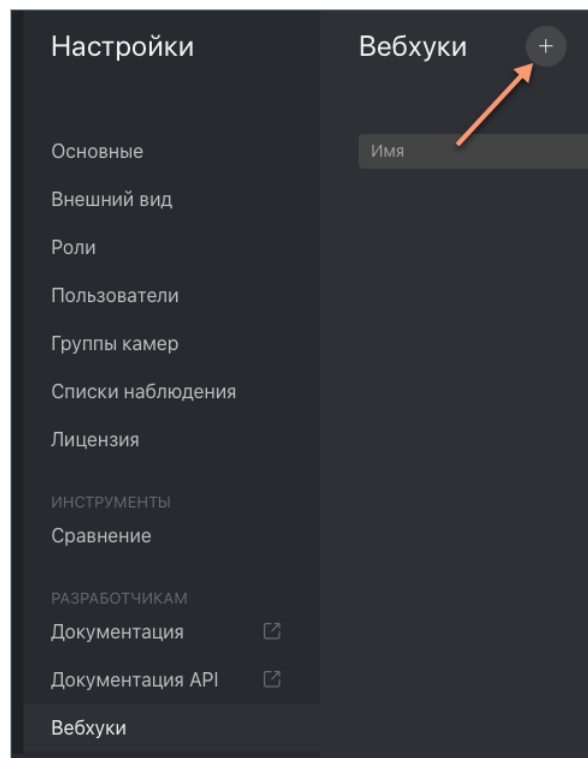
3.2.1 Настройка вебхука

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

Вебхук

• URL

<http://mywebhook.org/1>

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"
    ],
    "camera_group_in": [],
    "camera_in": []
  }
}
```

☒ Активный

Сохранить Назад

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 и показаниях счетчика, удовлетворяющих заданным фильтрам. Фильтровать события можно по следующим параметрам:

События распознавания (лицо, силуэт, автомобиль):

- `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_dossier_in`: id совпавшего досье, число.
- `matched`: статус события совпадение (`true` или `false`), логический.
- `confidence_gte`: минимальное значение уверенности алгоритма, число.

Эпизоды:

- `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.
- `matched`: статус события совпадение (`true` или `false`), логический.
- `events_count_gte`: минимальное количество событий в эпизоде, число.
- `events_count_lte`: максимальное количество событий в эпизоде, число.

Счетчики:

- `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.

```
{
  "face_events": {
    "allowed_bs_types": [
      "overall",
      "realtime"
    ],
    "camera_group_in": [],
    "camera_in": [],
    "matched_lists_in": [],
    "matched_dossier_in": [],
    "matched": true,
```

(continues on next page)

(продолжение с предыдущей страницы)

```

        "confidence_gte": 0.75
    },
    "body_events": {
        "allowed_bs_types": [
            "overall",
            "realtime"
        ],
        "camera_group_in": [],
        "camera_in": [],
        "matched_lists_in": [],
        "matched_dossier_in": [],
        "matched": true,
        "confidence_gte": 0.75
    },
    "car_events": {
        "allowed_bs_types": [
            "overall",
            "realtime"
        ],
        "camera_group_in": [],
        "camera_in": [],
        "matched_lists_in": [],
        "matched_dossier_in": [],
        "matched": true,
        "confidence_gte": 0.75
    },
    "episodes": {
        "allowed_types": [
            "episode_open",
            "episode_event",
            "episode_close"
        ],
        "camera_group_in": [],
        "camera_in": [],
        "matched_lists_in": [],
        "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
    }
}

```

Важно: Используйте только фильтры, соответствующие цели поиска. Для выключения фильтра удалите его из вебхука. Не оставляйте фильтр пустым ([]), поскольку в этом случае фильтр

вернет пустой результат.

Примечание: Для получения всех оповещений передайте скобки без вложенных фильтров:

```
{}
```

Совет: Пример №1. Получение оповещений обо всех событиях:

```
{ "events": {} }
```

Пример №2. Получение оповещений об открытии эпизодов с совпадениями в досье:

```
{ "episodes": { "allowed_types": ["episode_open"], "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 группы камер 1 или 3, если совпало досье с id 12 или 25.

```
{
    "events": {
        "camera_group_in": [1, 3],
        "matched_dossier_in": [12, 25],
    },
}
```

8. Поставьте флажок *Активный*.

9. Нажмите *Сохранить*.

3.2.2 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)])
```

(continues on next page)

(продолжение с предыдущей страницы)

```
# 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 о каждом произошедшем в системе событии, эпизоде и записи счетчика. Оповещения отправляются в следующем формате:

- Событие распознавания лица
- Событие распознавания силуэта
- Событие распознавания автомобиля
- Открытие эпизода
- Закрытие эпизода
- Запись счетчика

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
↳[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.
```

3.2.3 Подробные данные в вебхуках (Verbose)

По умолчанию оповещения по вебхуку содержат только идентификаторы таких сущностей, как досье, 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 dossiers, dossier-lists, camera and camera groups in webhooks
    'VERBOSE_WEBHOOKS': True,
    ...
}
...
```

В подробном режиме оповещения по вебхуку имеют следующий формат:

- Событие распознавание лица (verbose)

- Событие распознавание силуэта (verbose)
- Событие распознавание автомобиля (verbose)
- Открытие эпизода (verbose)
- Закрытие эпизода (verbose)
- Запись счетчика (verbose)

3.3 Интеграции с партнерами

3.3.1 Genetec Security Center

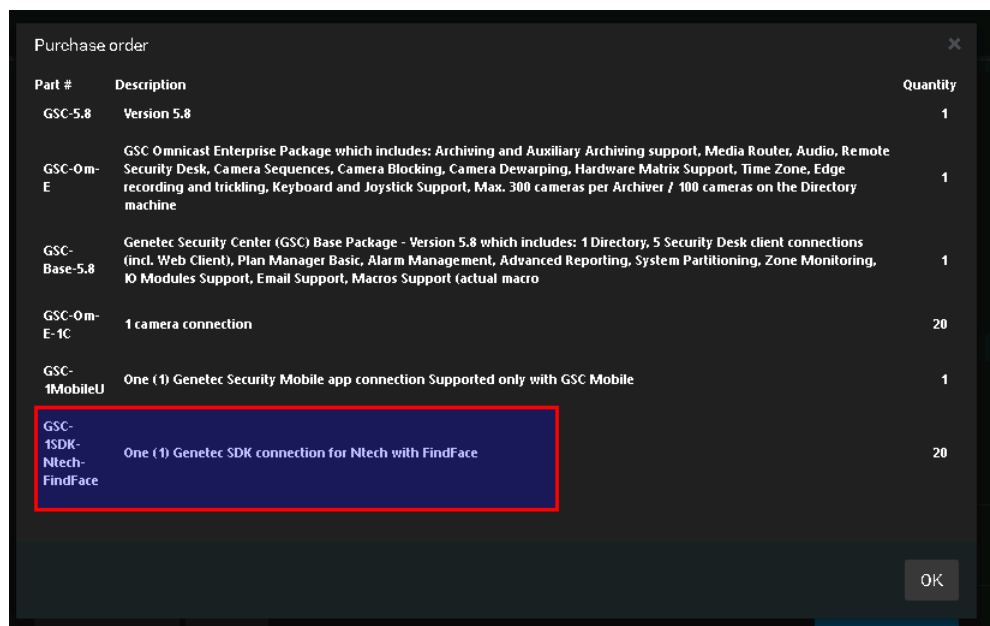
Интеграция FindFace Multi с программным комплексом Genetec Security Center позволяет добавлять функционал распознавания лиц в системы безопасности на базе Genetec.

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.

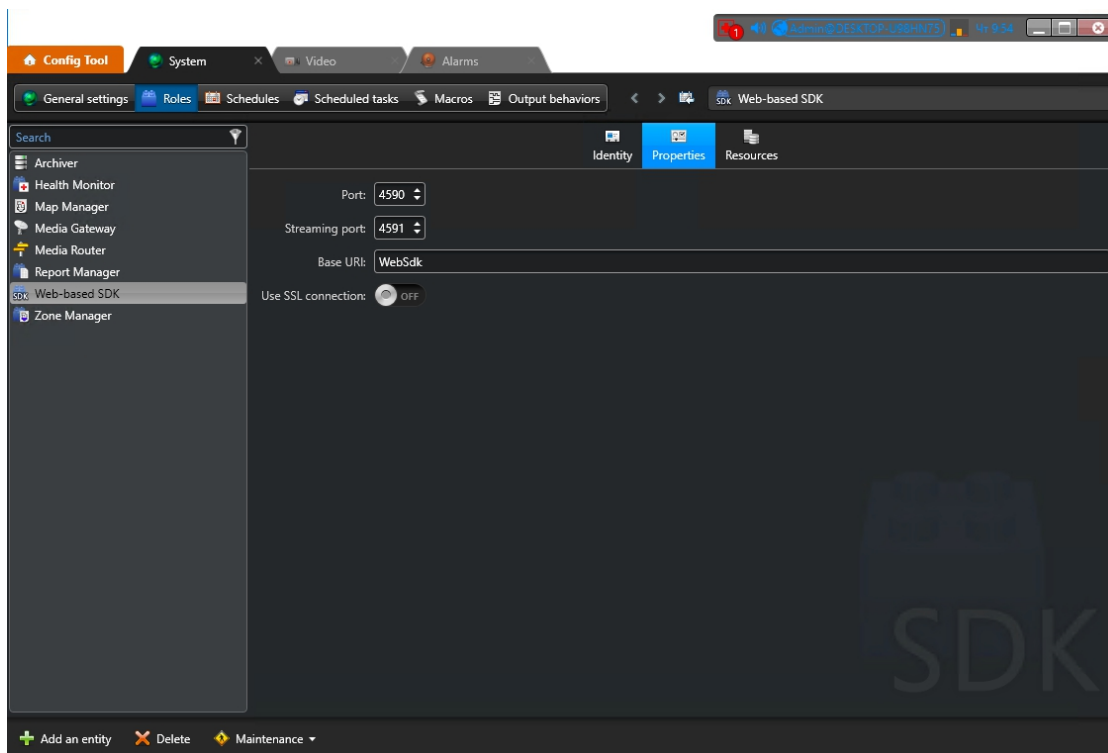


В этой главе:

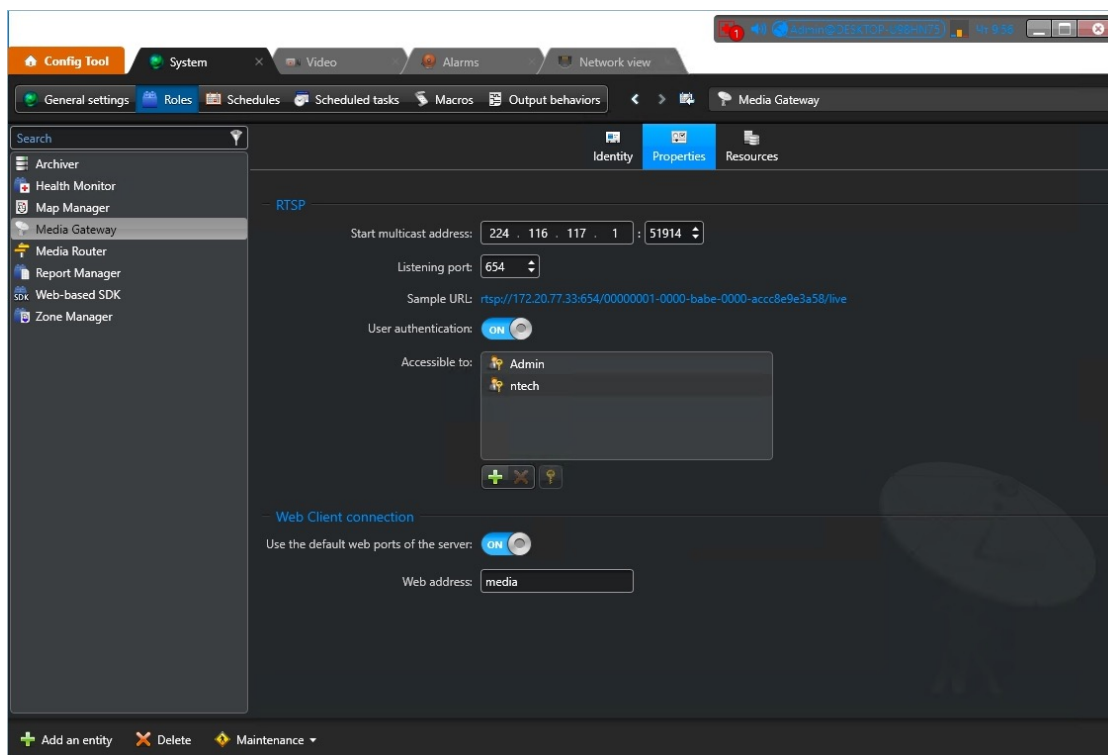
- *Настройка Genetec Web SDK и Media Gateway*
- *Создание оповещения в Genetec Security Center*
- *Включение интеграции Genetec в FindFace Multi*
- *Настройка точек доступа в FindFace Multi*
- *Импорт камер из Genetec Security Center*
- *Создание списков наблюдения и досье в FindFace Multi*

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*.



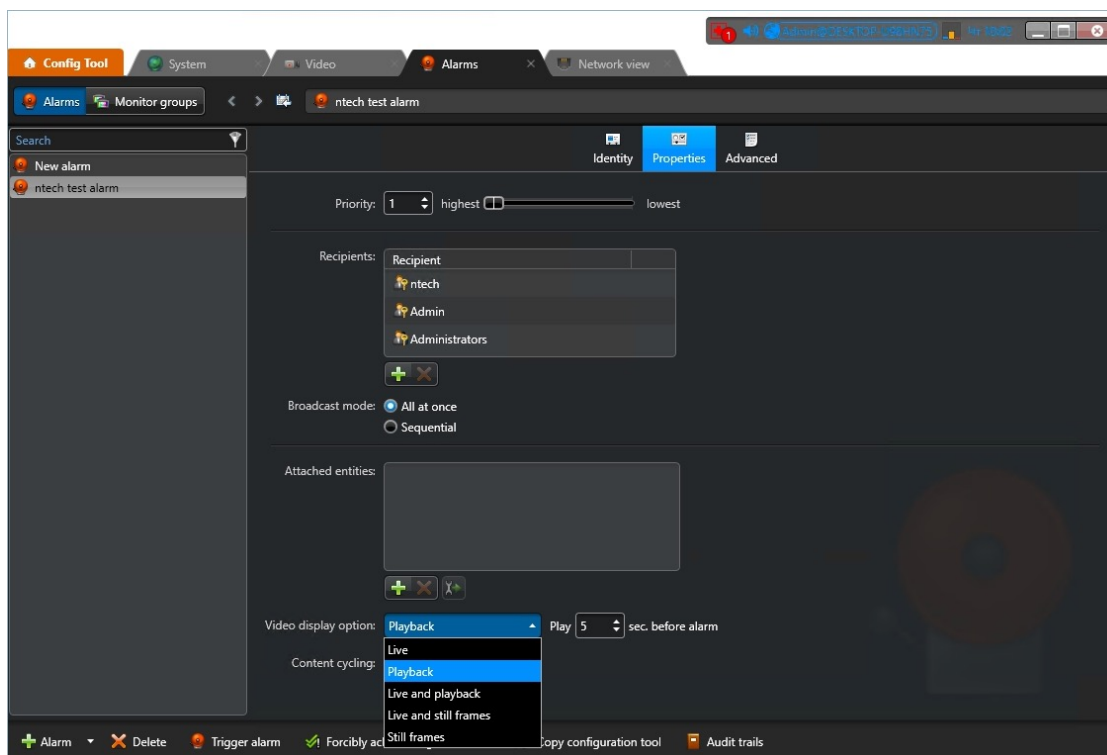
Для того чтобы развернуть Media Gateway в Genetec Config Tool, ознакомьтесь с содержанием главы *Security Center Administrator Guide -> Chapter 24: Video Deployment*.



Важно: Убедитесь, что фаервол настроен таким образом, что порты WebSDK и Media Gateway остаются открытыми.

Создание оповещения в Genetec Security Center

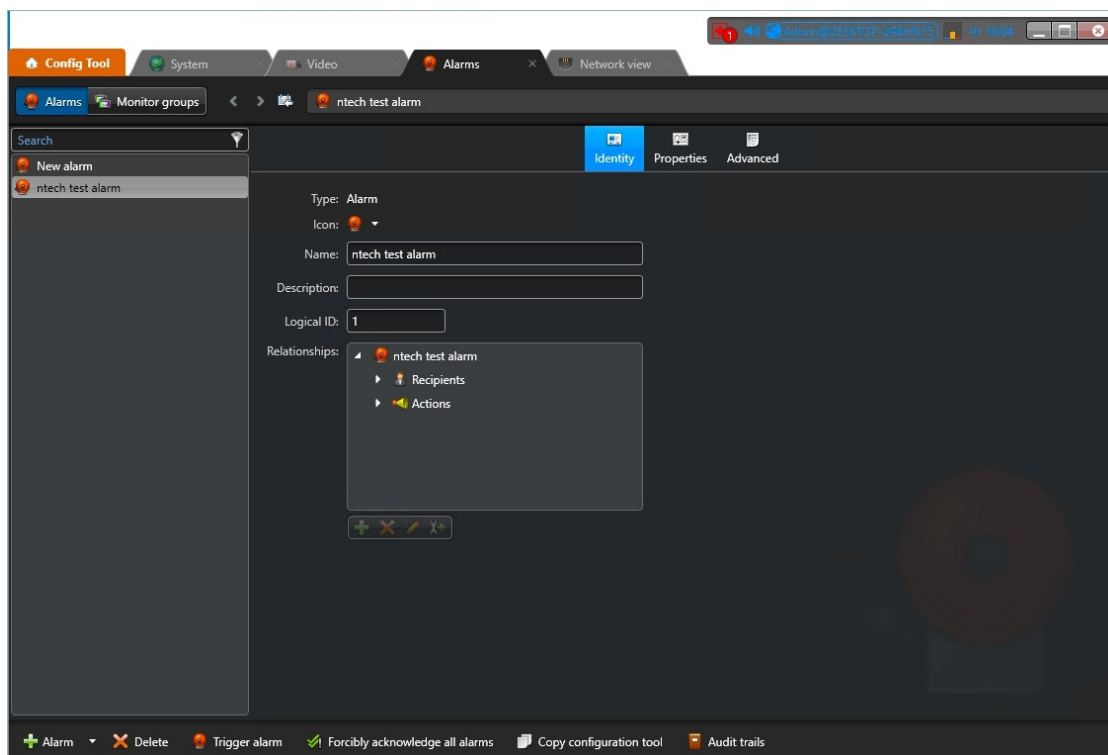
Создайте и настройте новое оповещение **Alarm** в Genetec Config Tool, руководствуясь разделом *Security Center Administrator Guide -> Chapter 48: Alarms -> Creating Alarms*.



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
```

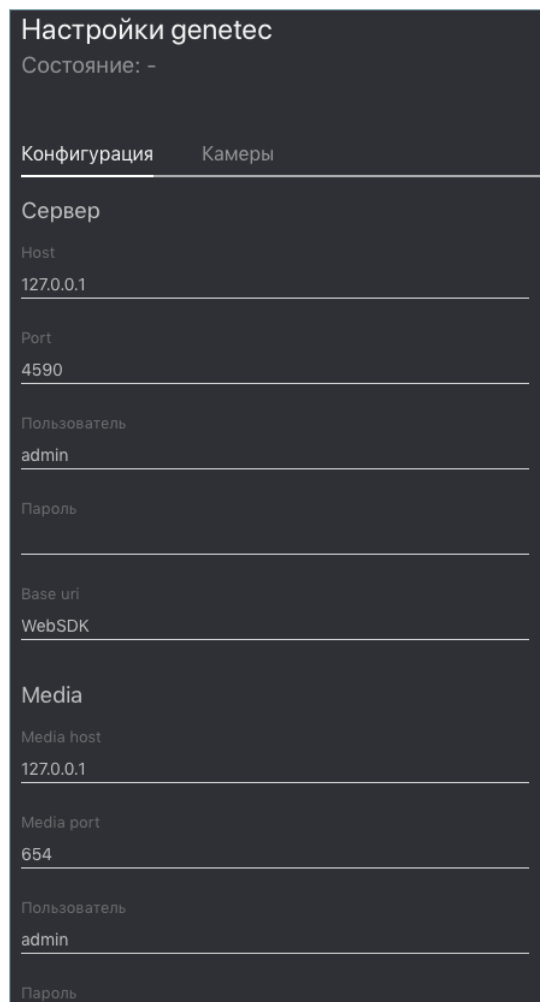
3. Перезапустите findface-security.

```
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.

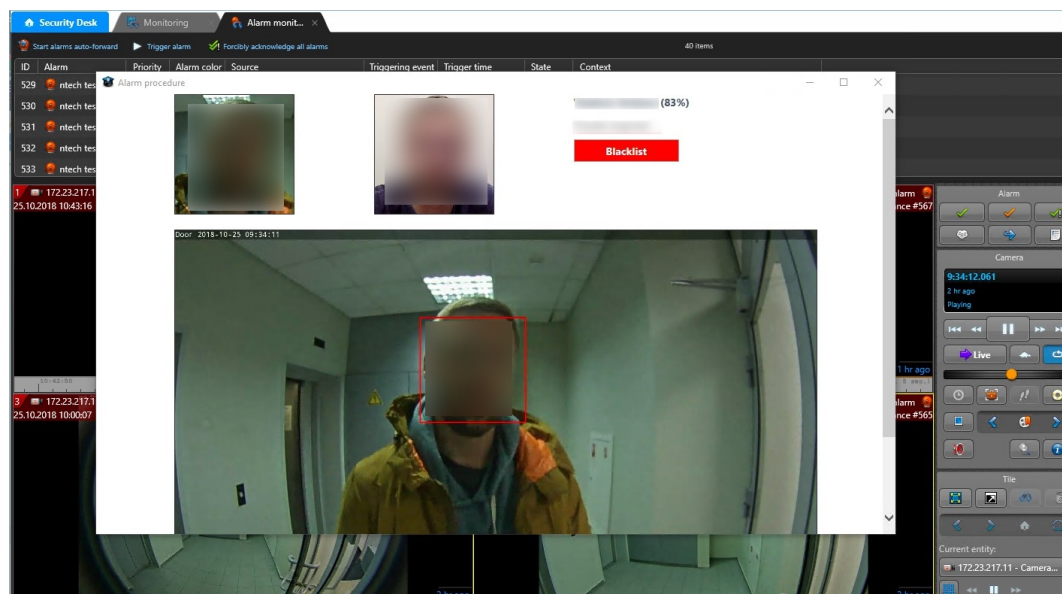
Создание списков наблюдения и досье в FindFace Multi

После настройки точек доступа и импорта камер завершите интеграцию, создав *базу данных досье*. 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 найдено досье, 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 базы данных досье
- имя человека и комментарий из досье
- 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.

3.3.2 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 лиц из досье.

Важно: Один экземпляр FindFace Multi может взаимодействовать только с одним сервером Аххон Next.

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 `FFSECURITY->AXXON`. Заполните ее так, как показано в примере ниже. В параметре `api` укажите адрес сервера 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.

```
FFSECURITY['AXXON'] = [
    {
        'name': 'server_name',
        'api': 'http://example.com/',
        'rtsp': 'rtsp://example.com:554/',
        'user': 'user',
        'password': 'password',
    }
]
```

3. (Опционально). Если в событиях распознавания лиц требуется отображать клипы видео из Аххон Next, раскомментируйте секцию `FFSECURITY_UI_CONFIG['dossier']`.

```
FFSECURITY_UI_CONFIG['dossier'] = {
    'video': True,
}
```

4. Создайте камеры в FindFace Multi (см. *Управление видеокameraми*). При создании камер вам потребуется ввести их URL в формате `axxon:<friendlyNameLong>`, где `friendlyNameLong` - имя камеры на сервере Аххон Next. Данное имя можно посмотреть в интерфейсе Аххон, или через API Аххон, выполнив команду:

```
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 единственной камеры из примера выше URL должен быть задан как `axxon:vhod_1.Vhod_1`.

На этом настройка интеграции будет завершена. Если интеграция настроена корректно, FindFace Multi будет выполнять проверку наличия лиц из досье в видеопотоках Аххон Next, а в событиях распознавания лиц будут отображаться клипы видео из Аххон Next (при соответствующих настройках).

3.3.3 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.

```
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
}

...
FFSECURITY = {

    # -- Optional parameters --

    # Edit CUSTOM_FIELDS section to customize dossier content.
    # Below is an example for integration FindFace Security with Sigur.

    'CUSTOM_FIELDS': {
        'dossier_meta': {
            'items': [
                {
                    'name': 'personid',
                    'default': '',
                    'label': 'PersonID',
                    'display': ['list', 'form'],
                    'description': 'Sigur person ID'
                },
                {
                    'name': 'firstname',
```

(continues on next page)

(продолжение с предыдущей страницы)

```

        'default': '',
        'label': 'First Name',
        'display': ['list', 'form'],
        'description': 'Sigur first name'
    },
    {
        'name': 'lastname',
        'default': '',
        'label': 'Last Name',
        'display': ['list', 'form'],
        'description': 'Sigur last name'
    },
    {
        'name': 'version',
        'default': '',
        'label': 'Version',
        'display': ['list', 'form'],
        'description': 'Sigur photo version'
    }
],
'filters': [
    {
        'name': 'personid',
        'label': 'Sigur person ID filter',
        'field': 'personid'
    }
]
},

```

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. 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_1.0.60.1-0_all.deb
wget https://sigur.com/dl/spnxserver_1.0.60.1-0_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

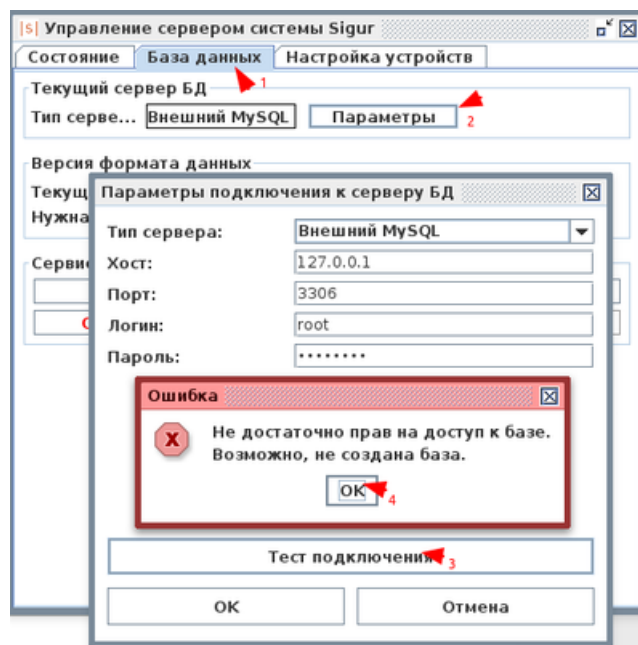
Для настройки и запуска Сервера Sigur выполните следующие действия:

1. Откройте панель администратора Сервера Sigur.

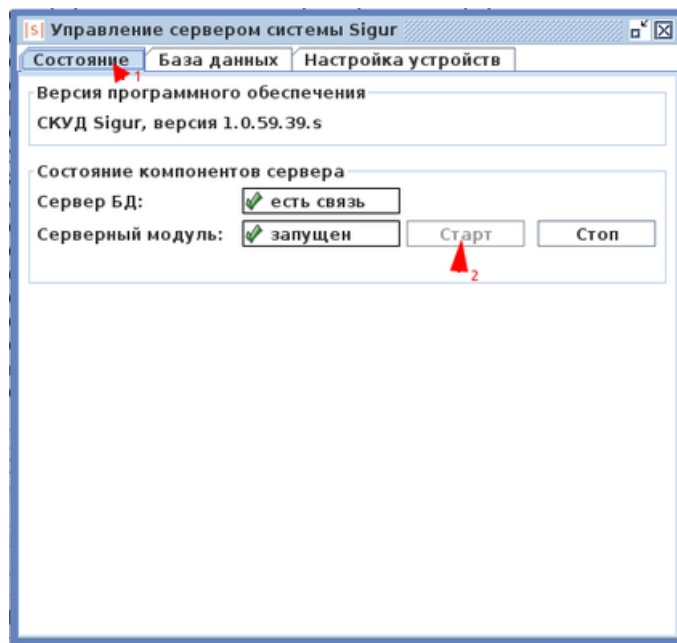
```
sudo spnxadmin
```

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

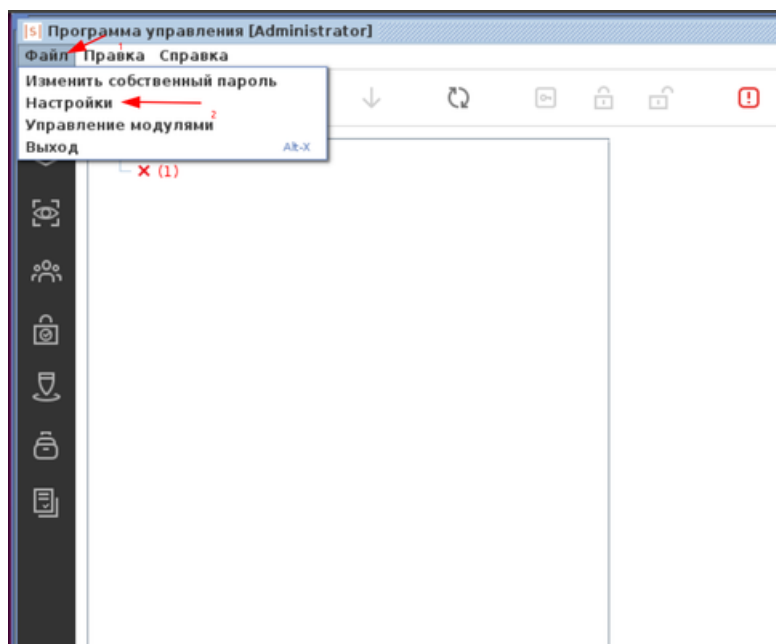
Для настройки Клиента Sigur выполните следующие действия:

1. Откройте панель клиента Sigur.

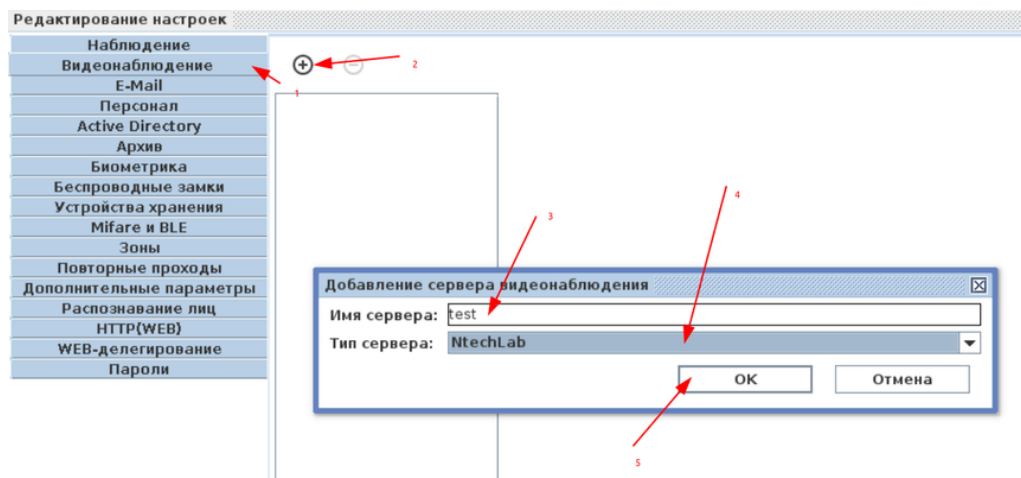
```
spxclient
```

2. Добавьте сервер FindFace Multi в Sigur:

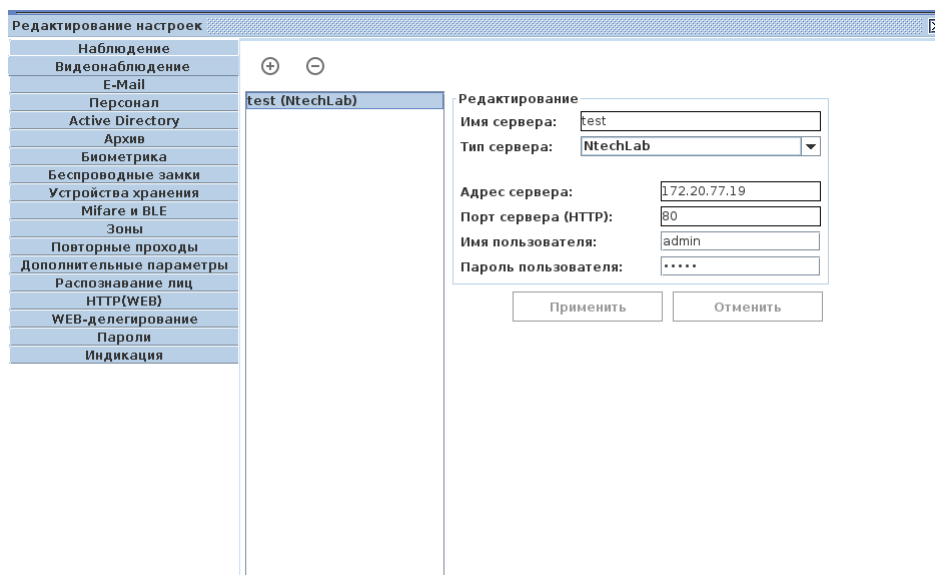
1. В главном меню перейдите *Файл -> Настройки -> Видеонаблюдение*.



2. В открывшемся окне выберите поставщика *NtechLab* и введите произвольное имя сервера.

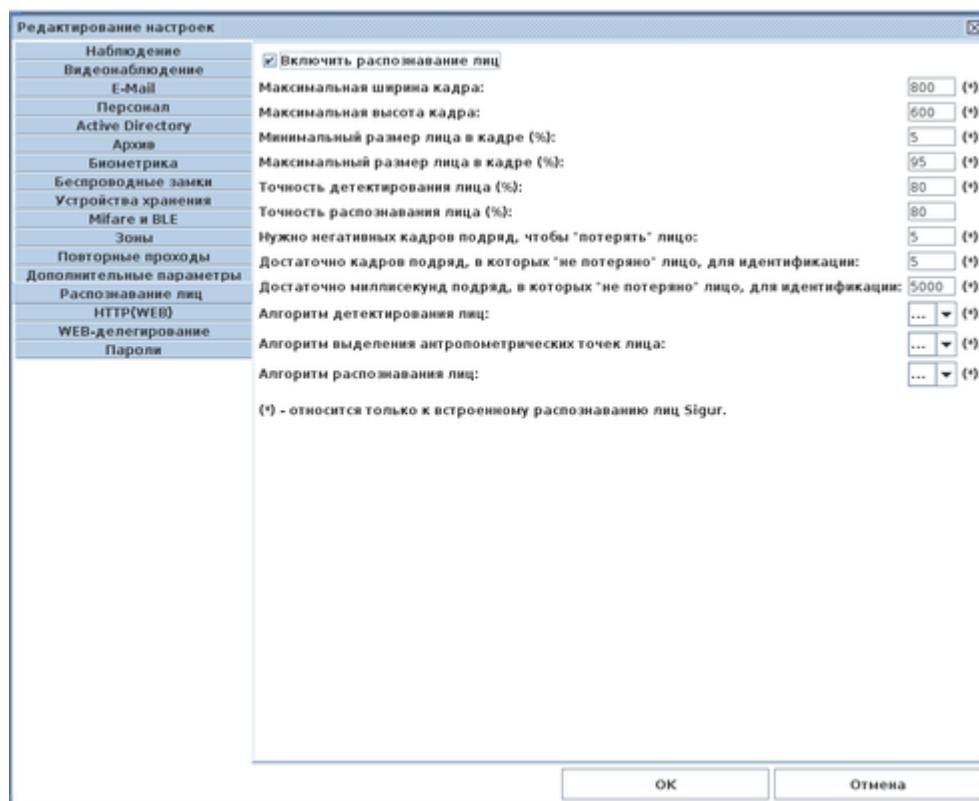


3. После того как в списке появится сервер, введите настройки доступа, которые вы указали в файле конфигурации *findface-security* (*логин и пароль* в блоке плагина Sigur). Порт для доступа Sigur к серверу FindFace Multi - 80.



4. Нажмите *Применить*.
3. Перейдите на вкладку *Распознавание лиц* и поставьте флажок *Включить распознавание лиц*.

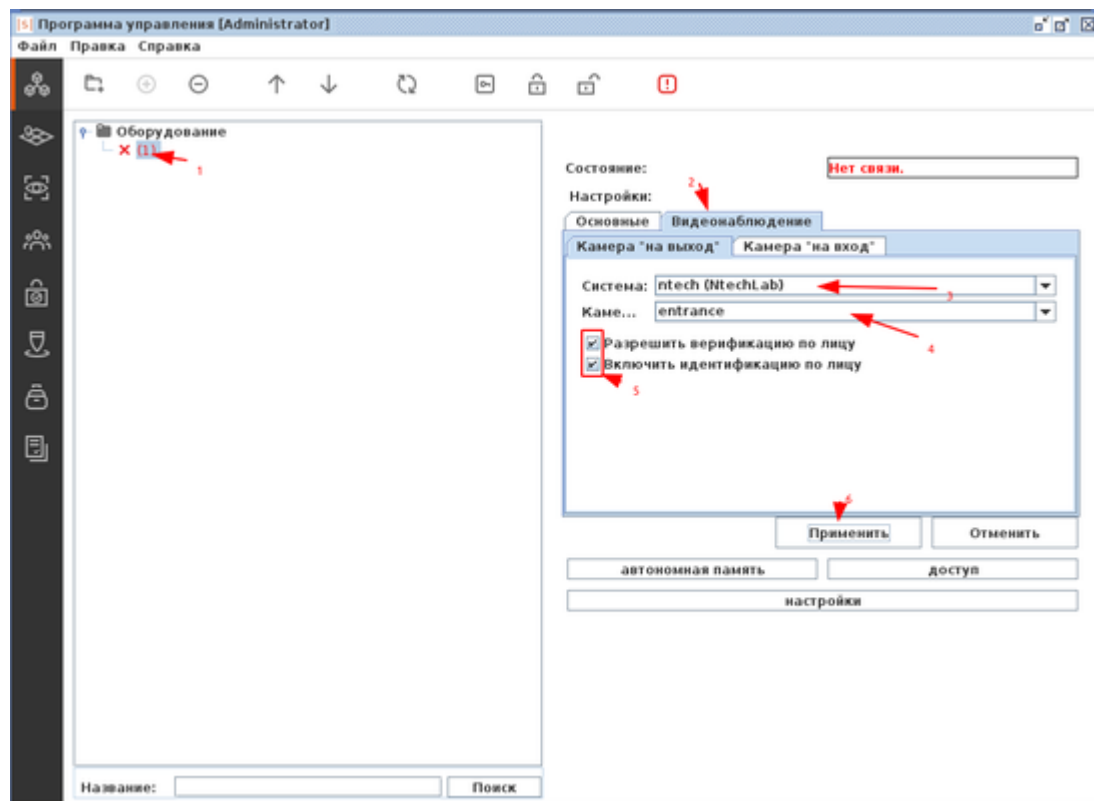
Важно: Значение в поле *Точность распознавания лица (%)* рекомендуется устанавливать ниже или равным установленному в *настройках FindFace Multi*.



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*.

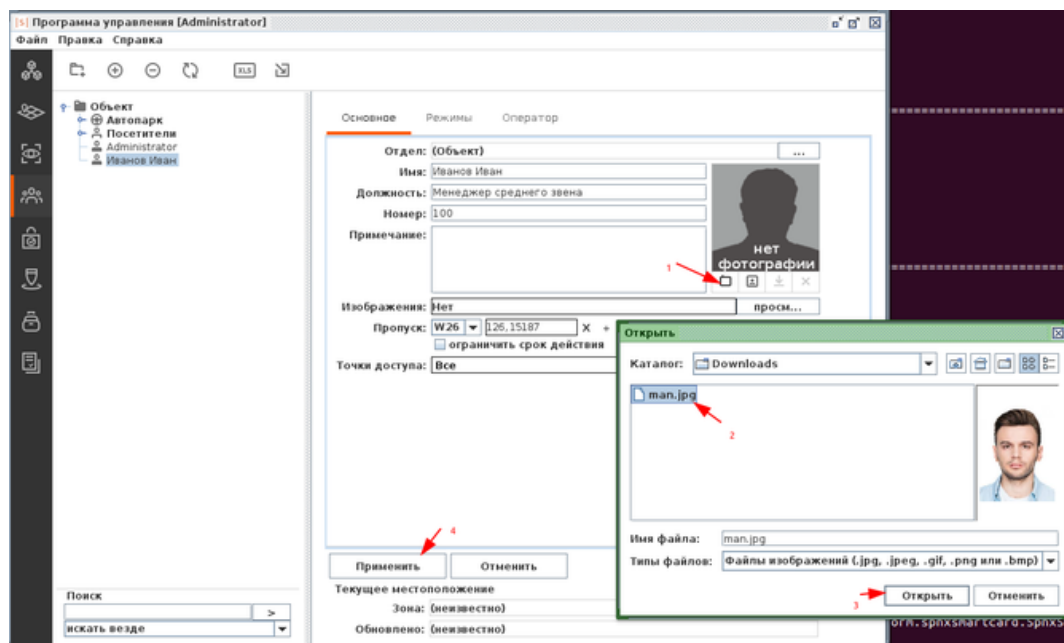


3. Установите следующие настройки:
 - Система: NtechLab.
 - Камера: в соответствии с камерой на данной точке входа.
 - флажки *Разрешить верификацию по лицу* и *Разрешить идентификацию по лицу* установлены.
4. Нажмите *Применить*.

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.

f

`facrouter.plugin`, [88](#)

n

`ntech.sfapi_client.client`, [91](#)

`ntech.sfapi_client.filters`, [94](#)

`ntech.sfapi_client.gallery`, [92](#)

O

`objects`, [90](#)

СИМВОЛЫ

- `__init__()` (метод `ntech.sfapi_client.filters.Detection`), 96
- `__init__()` (метод `ntech.sfapi_client.filters.Face`), 97
- ### А
- `add()` (метод `ntech.sfapi_client.gallery.Gallery`), 92
- ### В
- `BBox` (класс в `objects`), 90
- ### С
- `Client` (класс в `ntech.sfapi_client.client`), 91
- `create()` (метод `ntech.sfapi_client.gallery.Gallery`), 93
- ### Д
- `delete()` (метод `ntech.sfapi_client.gallery.Gallery`), 93
- `detect()` (метод `ntech.sfapi_client.client.Client`), 91
- `Detection` (класс в `ntech.sfapi_client.filters`), 96
- `drop()` (метод `ntech.sfapi_client.gallery.Gallery`), 94
- ### Е
- `Face` (класс в `ntech.sfapi_client.filters`), 97
- `facrouter.plugin`
модуль, 88
- `Filter` (класс в `ntech.sfapi_client.filters`), 94
- ### Г
- `Gallery` (класс в `ntech.sfapi_client.gallery`), 92
- `gallery()` (метод `ntech.sfapi_client.client.Client`), 91
- `get()` (метод `ntech.sfapi_client.gallery.Gallery`), 93
- `gte()` (метод класса `ntech.sfapi_client.filters.Id`), 95
- `gte()` (метод (метод класса `ntech.sfapi_client.filters.Meta`), 96
- ### И
- `Id` (класс в `ntech.sfapi_client.filters`), 95
- ### Л
- `list()` (метод `ntech.sfapi_client.gallery.Gallery`), 92
- `lte()` (метод класса `ntech.sfapi_client.filters.Id`), 95
- `lte()` (метод (метод класса `ntech.sfapi_client.filters.Meta`), 95
- ### М
- `Meta` (класс в `ntech.sfapi_client.filters`), 95
- ### Н
- `ntech.sfapi_client.client`
модуль, 91
- `ntech.sfapi_client.filters`
модуль, 94
- `ntech.sfapi_client.gallery`
модуль, 92
- ### О
- `objects`
модуль, 90
- `objects.DetectFace` (класс в `objects`), 90
- `objects.DetectResponse` (класс в `objects`), 90
- `objects.Face` (класс в `objects`), 90
- `objects.FaceId` (класс в `objects`), 90
- `objects.ListResponse` (класс в `objects`), 90
- `oneof()` (метод (метод класса `ntech.sfapi_client.filters.Id`), 95
- `oneof()` (метод (метод класса `ntech.sfapi_client.filters.Meta`), 96
- ### Р
- `Plugin` (класс в `facrouter.plugin`), 88

`preprocess()`, 86
`preprocess()` (*метод facerouter.plugin.Plugin*), 88
`process()`, 87
`process()` (*метод facerouter.plugin.Plugin*), 89

S

`serialize()` (*метод*
ntech.sfapi_client.filters.Filter), 94
`SfapiClient.SfapiMalformedResponseError`
(*класс в ntech.sfapi_client.filters*), 98
`SfapiClient.SfapiRemoteError` (*класс в*
ntech.sfapi_client.filters), 97
`shutdown()`, 88
`shutdown()` (*метод facerouter.plugin.Plugin*), 89
`subset()` (*метод класса*
ntech.sfapi_client.filters.Meta), 96

U

`update()` (*метод ntech.sfapi_client.gallery.Gallery*),
94

модуль

`facerouter.plugin`, 88
`ntech.sfapi_client.client`, 91
`ntech.sfapi_client.filters`, 94
`ntech.sfapi_client.gallery`, 92
`objects`, 90