
Программный комплекс анализа и распознавания биометрических данных FindFace

НТех лаб

нояб. 22, 2024

1	Руководство системного администратора	3
1.1	Архитектура	3
1.2	Системные требования	9
1.3	Требования к камерам видеонаблюдения	11
1.4	Лицензирование	14
1.5	Развертывание ПК FindFace	15
1.6	Обслуживание и устранение неисправностей	51
1.7	Приложения	76
2	Руководство пользователя	133
2.1	Первые шаги после установки	133
2.2	Работа с ПК FindFace	137
2.3	Расширенный функционал	207
3	Руководство по интеграции	241
3.1	HTTP API	241
3.2	Вебхуки	242
3.3	Интеграции с партнерами	266
3.4	Пользовательские плагины	275
	Содержание модулей Python	293
	Алфавитный указатель	295

Программный комплекс анализа и распознавания биометрических данных FindFace (далее ПК FindFace) — это многофункциональная биометрическая система, в основе которой лежит FindFace Enterprise Server, передовая технология распознавания лиц на базе искусственного интеллекта. ПК FindFace 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.

ПК FindFace обнаруживает и идентифицирует человеческие лица и уведомляет ответственных лиц об их появлении. ПК FindFace также распознает такие атрибуты лица, как пол, возраст, эмоции, очки, наличие медицинской маски, борода и др., и отображает данную информацию для каждого обнаруженного на видеоизображении лица.

Интегрированная в ПК FindFace антиспуфинговая система гарантирует, что перед камерой находится живой человек, и исключает возможность мошенничества с использованием фотографии лица на бумаге или экране мобильного устройства.

ПК FindFace поддерживает интеграцию сторонних решений через *HTTP API*, *вебхуки* и *плагины*, так что вы с легкостью сможете усовершенствовать свою текущую систему или приложение, добавив в них функционал распознавания лиц.

Ключевые функции

- Платформа на основе AI.
- Быстрая и надежная биометрическая идентификация в реальном времени по базам данных досье.
- Поддержка потокового видео и архивов, большинства видеоформатов и кодеков, которые могут быть декодированы FFmpeg.
- Верификация лиц.
- AI-распознавание пола, возраста, эмоций, очков, бороды, медицинской маски и других атрибутов лица.
- AI-детектор живых лиц (Liveness).
- AI-распознавание персон.
- Поиск по базам данных.
- Возможность подсчета лиц и силуэтов на подключенных камерах.
- 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 и аналитику KYC.
- Детектор живых лиц (liveness) как автономный сервис.

Полезные мелочи

- Быстрое создание базы данных досье.
- Полная настройка досье.
- Поддержка дедупликации 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, вебхуки и плагины на python.
- Партнерские интеграции с популярными системами.

Руководство системного администратора

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

1.1 Архитектура

Хотя взаимодействие с ПК FindFace 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 ПК FindFace.

В этой главе:

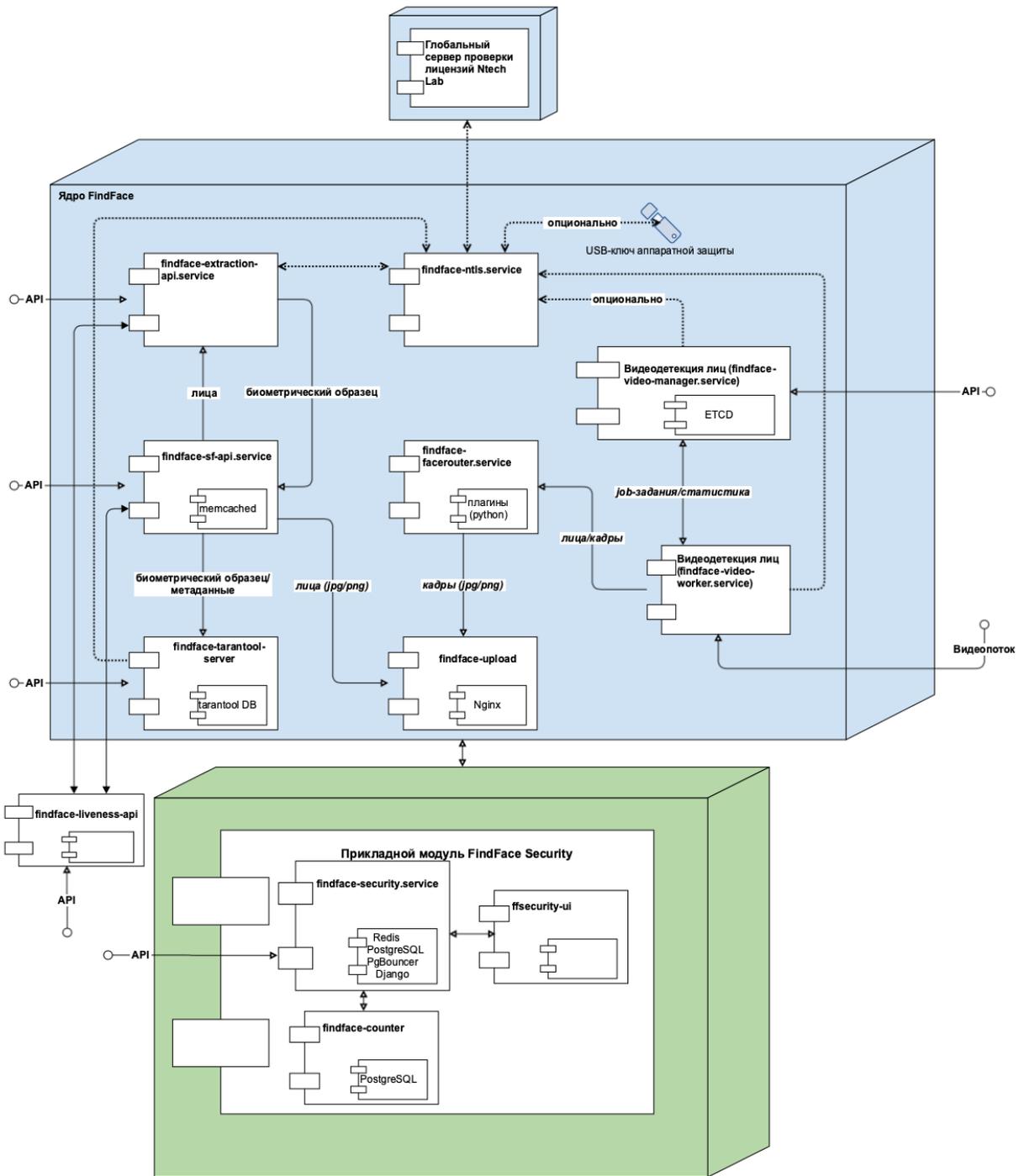
- *Архитектурные элементы*
 - *Схема архитектуры*
 - *Ядро FindFace*
 - *Прикладной модуль ПК FindFace*
- *Развертывание на одиночном сервере или в кластере*
- *Аппаратное ускорение на CPU и GPU*

1.1.1 Архитектурные элементы

ПК FindFace состоит из следующих основных архитектурных элементов:

- Ядро FindFace, передовая технология распознавания лиц на базе искусственного интеллекта, которая может использоваться в качестве отдельного продукта FindFace Enterprise Server.
- ПК FindFace, готовый прикладной модуль к FindFace Enterprise Server.

Схема архитектуры



Ядро FindFace

Ядро FindFace включает в себя следующие компоненты:

Программный комплекс анализа и распознавания биометрических данных FindFace

Компонент	Используемые порты	Описание	Поставщик
findface-extraction-api	8666	Сервис, использующийся для обнаружения лиц на статических изображениях и извлечения из лиц биометрических образцов (векторов признаков). Сервис также выполняет распознавание таких атрибутов лица, как пол, возраст, эмоции, борода, очки, медицинская маска и др., а также распознавание человеческих силуэтов (если были заданы соответствующие настройки). Работа на базе нейронных сетей. Поддерживается CPU- или GPU-ускорение.	Собственная разработка NtechLab
findface-sf-api	8411	Сервис, реализующий HTTP API обнаружения и распознания лиц.	
findface-tarantool-server	82001, порты шардов (по умолчанию 330xx, 81xx)	Сервис, обеспечивающий взаимодействие между сервисом findface-sf-api и биометрической базой данных (базой, в которой хранятся биометрические образцы) на основе Tarantool.	
findface-upload	8333	Веб-сервер на базе NginX, используемый как хранилище исходных изображений, миниатюр лиц и нормализованных изображений лиц.	
findface-facerouter	8820	Сервис, который используется для задания правил обработки обнаруженных на видео лиц. В ПК FindFace функции findface-facerouter выполняются findface-security (см. <i>Прикладной модуль ПК FindFace</i>). Однако в целях интеграции вы можете установить и настроить данный компонент (см. <i>Пользовательские плагины</i>).	
findface-video-manager	8810, 18811	Сервис, являющийся частью модуля видеодетекции лиц, через который осуществляется управление детекцией лиц на видео, а именно задаются настройки и список видеопотоков для обработки.	
findface-video-worker	8999	Сервис, часть модуля видеодетекции лиц, который распознает лица на видео и отправляет их нормализованные изображения, соответствующие видеокadres и метаданные (такие как ID камеры и время обнаружения лица) 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.	
findface-ntls	443 (TCP), 3133, 3185	Локальный сервер лицензий, который проверяет подлинность <i>лицензии</i> ПК FindFace, взаимодействуя с глобальным сервером лицензий NtechLab. Для закрытых систем поддерживается работа с аппаратными лицензионными ключами. Поддерживается лицензирование через прокси-сервер.	Tarantool
Tarantool	Порты шардов (по умолчанию 330xx, 81xx)	Стороннее программное обеспечение, на основе которого реализована биометрическая база данных, хранящая извлеченные биометрические образцы (векторы признаков) и события обнаружения лиц. Системные данные, досье, пользовательские аккаунты и настройки камер хранятся в PostgreSQL (часть прикладного модуля ПК FindFace).	
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 лиц.	etcd
NginX	80; SSL: 8002, 8003, 443	Стороннее программное обеспечение, которое реализует веб-интерфейсы системы.	nginx
1.1. Архитектура			7
memcached	11211	Стороннее программное обеспечение, реализующее сервис кэширования данных в оперативной памяти на основе хеш-таблицы. Используется для временного хранения извлеченных биометрических образцов перед их за-	memcached

Прикладной модуль ПК FindFace

Прикладной модуль ПК FindFace включает в себя следующие компоненты:

Компонент	Используемые порты	Описание	Поставщик
findface-security	Настраивается	Компонент, обеспечивающий доступ конечного пользователя к функциям ядра FindFace. Обеспечивает взаимодействие между ядром FindFace и веб-интерфейсом, функционирование системы в целом, HTTP и веб-сокеты, обновление биометрической базы данных, 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> .	Собственная разработка NtechLab
findface-security-ui	Настраивается	Основной веб-интерфейс, использующийся для взаимодействия с ПК FindFace. Позволяет работать с событиями идентификации лиц, искать лица 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 лиц.	
findface-liveness-api	8301	Помимо встроенного функционала, предоставляемого <code>findface-video-worker</code> , распознавание живых лиц (liveness) также предоставляется автономным сервисом <code>findface-liveness-api</code> . 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 персон с разбиением по категориям (спискам наблюдения), а также данные внутреннего характера, такие как профили пользователей ПК FindFace, настройки видеокамер и пр. Биометрические данные лиц и события идентификации лиц хранятся в Tarantool (часть ядра FindFace).	PostgreSQL
Pgbouncer	439	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 <code>findface-security</code> .	Redis
Django	5439	Стороннее программное обеспечение, реализующее веб-фреймворк для веб-интерфейса ПК FindFace.	Django

См. также:

Подробнее о компонентах

1.1.2 Развертывание на одиночном сервере или в кластере

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

- Центральный сервер ПК FindFace, взаимодействующий с несколькими дополнительными серверами для обработки видео (с одним установленным компонентом `findface-video-worker`).
- Полностью распределенная архитектура ПК FindFace. Может потребоваться балансировка нагрузки.

Подробнее см. раздел *Типичная установка в кластере*.

1.1.3 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 ПК FindFace 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. *Системные 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 Системные требования

Для расчета характеристик серверов ПК FindFace используйте приведенные ниже системные требования.

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

В этой главе:

- *Базовая конфигурация*
- *Необходимые навыки администратора*

1.2.1 Базовая конфигурация

Важно: Если разрешение используемой камеры превышает 1280x720 пикселей, настоятельно рекомендуется использовать пакет с ускорением на GPU `findface-video-worker-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 ПК FindFace 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 ПК FindFace 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 ПК FindFace требуется 15 Гб. Суммарный объем определяется в зависимости от требуемой глубины архива событий в базе данных и в логе из расчета 1.5 Мб на 1 событие	
Операционная система	Ubuntu 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.

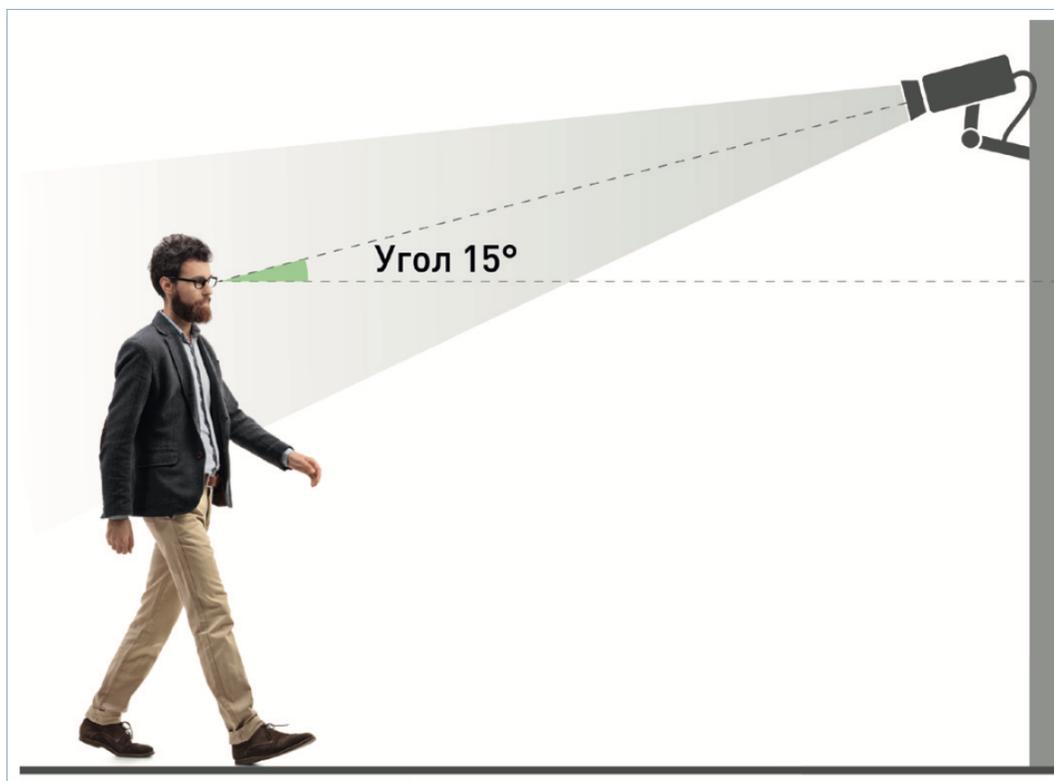
1.2.2 Необходимые навыки администратора

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

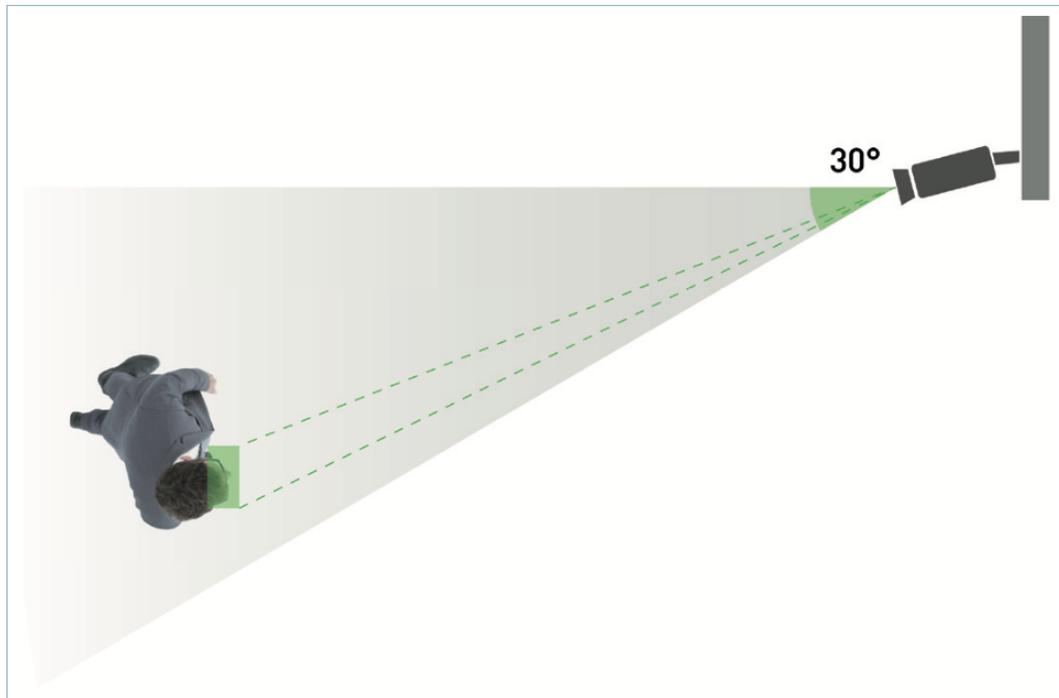
1.3 Требования к камерам видеонаблюдения

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

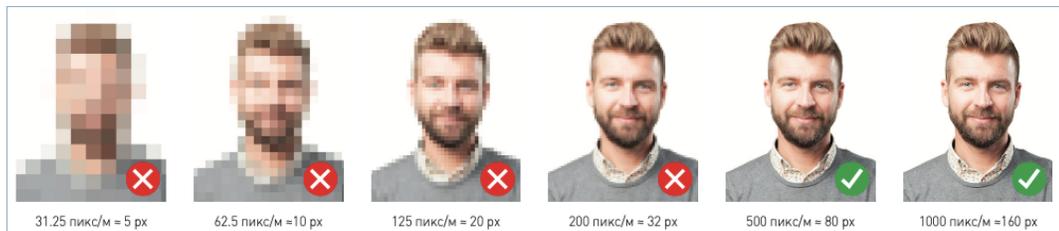
1. Для корректной детекции в видеопотоке установите камеру таким образом, чтобы в ее поле зрения обязательно появлялось лицо каждого человека, входящего в зону наблюдения.
2. Угол вертикального наклона видеокамеры не должен превышать 15° . Вертикальный наклон — это отклонение оптической оси видеокамеры 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.

1.4 Лицензирование

В этой главе:

- *Принцип лицензирования*
- *Просмотр и обновление лицензии*

1.4.1 Принцип лицензирования

ПК FindFace лицензируется по следующим критериям:

1. Количество извлекаемых биометрических образцов. Биометрические образцы извлекаются из лиц, обнаруженных на видео, из фотографий, загружаемых в досье, а также при построении т. н. центроидов в процессе *кластеризации персон*.

Схема лицензирования выглядит следующим образом:

- События: 1 событие распознавания лица на видео = 1 лицо в лицензии.
 - Досье: 1 фотография в досье = 1 лицо в лицензии.
 - Персоны: 1 персона = 1 лицо в лицензии.
2. Количество камер в системе.
 3. Количество экземпляров моделей, используемых `findface-extraction-api`.
 4. Распознавание атрибутов лица: пол/возраст/эмоции/очки/борода/медицинская маска.
 5. Распознавание живых лиц в реальном времени (Liveness).
 6. Интеграции с партнерами.

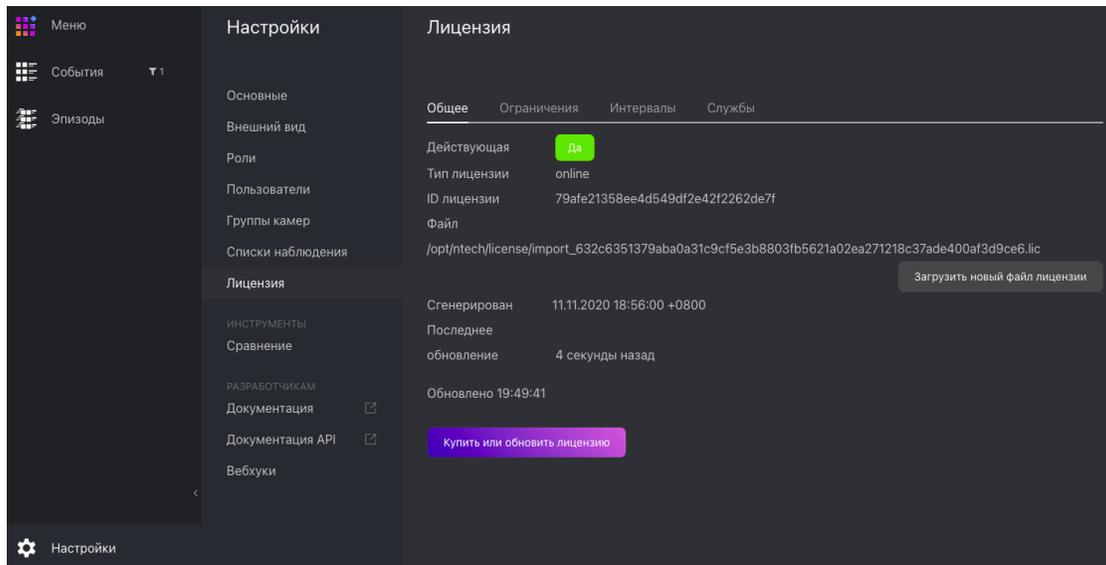
Вы можете выбрать между онлайн-лицензированием и лицензированием в закрытой сети:

- Онлайн-лицензирование выполняется через Глобальный менеджер лицензий NtechLab `license.ntechlab.com` и требует наличия стабильного интернет-соединения, DNS и открытый порт 443 TCP. После отключения от интернета система продолжит работать в автономном режиме в течение 4-х часов. 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.

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.

1.4.2 Просмотр и обновление лицензии

После установки ПК FindFace 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

1.5 Развертывание ПК FindFace

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

1.5.1 Развертывание из консольного инсталлятора

Для развертывания ПК FindFace 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 ПК FindFace из инсталлятора выполните следующие действия:

1. Загрузите файл инсталлятора `findface-security-and-server-4.5.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-4.5.run
```

4. Запустите файл `.run`.

```
sudo ./findface-security-and-server-4.5.run
```

Инсталлятор задаст вам несколько вопросов, после чего проверит, соответствует ли сервер системным требованиям. Вопросы следующие:

1. Устанавливаемый продукт: ПК FindFace.
2. Тип установки:
 - 1: установить ПК FindFace на одиночном физическом сервере.
 - 2: установить ПК FindFace 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 ПК FindFace на других серверах, не отвечая повторно на вопросы инсталлятора.

При выборе установки одиночного сервера ПК FindFace, 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 `jackfruit_480`. В противном случае вам будет предложено сделать выбор между `jackfruit_480` и предыдущей моделью. Категорически не рекомендуется использовать инсталлятор для обновления системы.

Сервис	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 ПК FindFace.
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 лиц и их атрибутов (пол, возраст, эмоции, очки, борода, медицинская маска и др.). Устанавливаются.
findface-gpudetector-data	Данные gpu-детектора NTechLab. Устанавливается.
python3-ntech.ffsecurity-client	Библиотека клиента Python, используемая в API ПК FindFace. 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 ПК FindFace в формате JSON.

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

Совет: Обязательно сохраните эти данные: они вам понадобятся.

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

5. Загрузите файл лицензии 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.

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

```
sudo ./findface-security-and-server-4.5.run -f /tmp/<findface-installer-*>.json
```

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

Важно: Для того чтобы сохранить совместимость ПК FindFace 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 Пошаговое развертывание из apt-репозитория

Данный раздел содержит подробную информацию о пошаговом развертывании компонентов ПК FindFace. 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 ПК FindFace и биометрической базы данных*

Установка apt-репозитория

Прежде всего установите apt-репозиторий FindFace следующим образом:

1. Загрузите файл инсталлятора `findface-security-and-server-4.5.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-4.5.run
```

4. Запустите файл `.run`.

```
sudo ./findface-security-and-server-4.5.run
```

Инсталлятор задаст вам несколько вопросов, после чего проверит, соответствует ли сервер системным требованиям. Вопросы следующие:

1. Устанавливаемый продукт: ПК FindFace.
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 ПК FindFace 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
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 ПК FindFace. Для лицензирования в закрытой сети вам также будет предоставлен ключ аппаратной защиты.

Лицензирование ПК FindFace обеспечивается следующим образом:

1. Разверните `findface-ntls`, сервер лицензий в составе ядра FindFace.

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

Совет: В файле конфигурации `/etc/findface-ntls.cfg` вы можете изменить папку для хранения файла лицензии и настроить удаленный доступ к веб-интерфейсу `findface-ntls`, используемому для управления лицензией. Подробнее см. *findface-ntls*.

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

2. Загрузите файл лицензии через веб-интерфейс `findface-ntls` одним из следующих способов:

- Откройте веб-интерфейс `findface-ntls`: `http://<NTLS_IP_address>:3185/#/`. Загрузите файл лицензии.

Совет: Впоследствии используйте основной веб-интерфейс ПК FindFace, чтобы посмотреть информацию о лицензиях, обновить или продлить лицензию (*Настройки -> Лицензия*).

- Непосредственно положите файл лицензии в предназначенную для этого папку (по умолчанию, `/opt/ntech/license`, может быть изменена в файле конфигурации `/etc/findface-ntls.cfg`).
3. При лицензировании в закрытой системе вставьте USB-ключ аппаратной защиты в USB-порт.
 4. Если лицензируемые компоненты установлены на удаленных серверах, впоследствии укажите IP-адрес сервера `findface-ntls` в их файлах конфигурации. Подробнее см. *findface-extraction-api*, *findface-tarantool-server*, *Видеодетекция лиц: findface-video-manager* и *findface-video-worker*.

См.также:

Лицензирование

Развертывание основной базы данных

Основная база данных ПК FindFace 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 ПК FindFace 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
...
```

4. В файле конфигурации `/etc/findface-extraction-api.ini` включите модели распознавания атрибутов лица, таких как пол, возраст, эмоции, очки, борода и медицинская маска, в зависимости от ваших нужд. 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 CPU-, так и GPU-модели. Подробнее см. *Распознавание атрибутов лица*.

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

models:
  age: faceattr/age.v1.cpu.fnk
  emotions: faceattr/emotions.v1.cpu.fnk
  face: face/jackfruit_480.cpu.fnk
  gender: faceattr/gender.v2.cpu.fnk
  beard: faceattr/beard.v0.cpu.fnk
  glasses3: faceattr/glasses3.v0.cpu.fnk
  medmask3: faceattr/medmask3.v2.cpu.fnk
```

Доступны следующие модели:

Атрибут лица	Ускорение	Параметр в файле конфигурации
биометрия лица	CPU	face: face/ifruit_320.cpu.fnk face: face/jackfruit_160.cpu.fnk face: face/jackfruit_320.cpu.fnk face: face/jackfruit_480.cpu.fnk
	GPU	face: face/ifruit_320.gpu.fnk face: face/jackfruit_160.gpu.fnk face: face/jackfruit_320.gpu.fnk face: face/jackfruit_480.gpu.fnk
возраст	CPU	age: faceattr/age.v1.cpu.fnk
	GPU	age: faceattr/age.v1.gpu.fnk
пол	CPU	gender: faceattr/gender.v2.cpu.fnk
	GPU	gender: faceattr/gender.v2.gpu.fnk
эмоции	CPU	emotions: faceattr/emotions.v1.cpu.fnk
	GPU	emotions: faceattr/emotions.v1.gpu.fnk
очки	CPU	glasses3: faceattr/glasses3.v0.cpu.fnk
	GPU	glasses3: faceattr/glasses3.v0.gpu.fnk
борода	CPU	beard: faceattr/beard.v0.cpu.fnk
	GPU	beard: faceattr/beard.v0.gpu.fnk
медицинская маска	CPU	medmask3: faceattr/medmask3.v2.cpu.fnk
	GPU	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.

```
models:
  gender: ""
  age: ""
  emotions: ""
```

- 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 webservice port, 0=disabled
## type:number env:CFG_STREAMER_PORT longopt:--streamer-port
port = 18999
## streamer url - how to access this worker on streamer_port
```

(continues on next page)

(продолжение с предыдущей страницы)

```
## type:string env:CFG_STREAMER_URL longopt:--streamer-url
url = 127.0.0.1:18999
```

6. 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 ПК FindFace и биометрической базы данных

Для развертывания прикладного модуля ПК FindFace, выполните следующие действия:

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

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

2. Перенесите схему базы данных из ПК FindFace 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 биометрической базы данных на основе Tarantool.

```
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/
...
```

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:

- `SERVICE_EXTERNAL_ADDRESS`: IP-адрес или URL ПК FindFace, 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`.
- `EXTERNAL_ADDRESS`: (Опционально) IP-адрес или URL, который используется для доступа к веб-интерфейсу ПК FindFace. Если параметр не задан, система автоматически определяет его как внешний IP-адрес. Для доступа в ПК FindFace 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 переадресации* с HTTP на HTTPS или вообще не указывайте его, оставив только IP-адрес.
- `SF_API_ADDRESS`: IP-адрес сервера `findface-sf-api`.
- `DATABASES` (раздел): заполните его следующим образом: `'PORT': 5439, 'USER': 'ntech', 'PASSWORD': '<пароль из /etc/pgbouncer/userlist.txt>'` (см. *Установка необходимого стороннего ПО*).

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

Совет: При необходимости установите 'IGNORE_UNMATCHED': True, чтобы отключить запись события в базу данных, если 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.

Важно: Если вы включили модели распознавания в файле конфигурации /etc/findface-extraction-api.ini, добавьте строку EVENTS_FEATURES в секцию FFSECURITY, в зависимости от того, какие модели были включены. Данная строка должна быть расположена между строками SF_API_ADDRESS и LIVENESS_THRESHOLD, как показано в примере ниже. Подробнее см. *Распознавание атрибутов лица*.

```
'EVENTS_FEATURES': ['gender', 'age', 'emotions', 'beard', 'glasses', 'medmask'],
```

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

# =====
# FindFace Security configuration file
# =====
#
# This config file is written in Python's syntax and interpreted at FindFace
↳ Security
# service startup. You have to restart the service in order to apply changes.
#
# If you have any questions or suggestions, please contact us at support@ntechlab.
↳ com

# =====
# GENERAL SETTINGS
# =====

# enables additional logs
DEBUG = False

# media files directory
MEDIA_ROOT = "/var/lib/findface-security/uploads"

# static files directory
STATIC_ROOT = "/var/lib/findface-security/static"

# language code
LANGUAGE_CODE = 'en-us'
```

(continues on next page)

(продолжение с предыдущей страницы)

```

# time zone
TIME_ZONE = 'UTC'

# 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': 'ZnUqeWKMzT6T2Bj2G4gbFn7cxRSMcxS7
→ '
    }
}

# Signature key for session encryption
# Use pugen -sncy 50 1/tr "" "." to generate your own unique key
SECRET_KEY = '8b26839acde2633362bdb176e741a650'

# =====
# FINDFACE SECURITY SETTINGS
# =====

# 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://172.20.77.120'

# EXTERNAL_ADDRESS is used to access objects created inside FFSecurity via external
→ links.
EXTERNAL_ADDRESS = ''

# - Base FFSecurity settings -

# enable permissions system
ENABLE_ACL = True

FFSECURITY = {
    # findface-video-worker authorization token
    'VIDEO_DETECTOR_TOKEN': '3243a92b03c3411d4faa3cdd72f967b6',

    # base face matching confidence threshold
    'CONFIDENCE_THRESHOLD': 0.745,

    # episodes specific matching threshold that is used to join faces in an episode
    'EPISODES_THRESHOLD': 0.689,

    # minimum face quality sufficient to add it to a dossier
    'MINIMUM_DOSSIER_QUALITY': 0.45,

```

(continues on next page)

(продолжение с предыдущей страницы)

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

# 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
'BLURRED_FULLFRAME_JPEG_QUALITY': 85,

# matched events older than EVENTS_MAX_MATCHED_AGE will be automatically
# deleted (every night at 1:17 am by default)
'EVENTS_MAX_MATCHED_AGE': 30,

# same as above but for unmatched events
'EVENTS_MAX_UNMATCHED_AGE': 30,

# same as EVENTS_MAX_MATCHED_AGE but for matched full frame images only
→ (thumbnails won't be deleted)
'EVENTS_MAX_FULLFRAME_UNMATCHED_AGE': 30,

# same as above but for unmatched full frame images only (thumbnails won't be
→ deleted)
'EVENTS_MAX_FULLFRAME_MATCHED_AGE': 30,

# same as above but for counter records
'COUNTER_RECORDS_MAX_AGE': 30,

# same as above but for person events (if no person events left in person, it
→ is deleted too)
'PERSON_EVENTS_MAX_AGE': 90,

# when closing episode, delete all events except the best episode event
'EPISODE_KEEP_ONLY_BEST_EVENT': False,

# NTLs licence server url
'NTLS_HTTP_URL': 'http://127.0.0.1:3185',

# 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',

# send serialized dossiers, dossier-lists, camera and camera groups in webhooks
'VERBOSE_WEBHOOKS': False,

# jpeg quality used when saving thumbnails
'THUMBNAIL_JPEG_QUALITY': 75,

# FFServer services urls
'VIDEO_MANAGER_ADDRESS': 'http://127.0.0.1:18810',
'SF_API_ADDRESS': 'http://127.0.0.1:18411',
```

(continues on next page)

(продолжение с предыдущей страницы)

```
'FFCOUNTER_ADDRESS': 'http://127.0.0.1:18300',
'LIVENESS_API_ADDRESS': 'http://127.0.0.1:18301',

# upload video archives to this path, it differs from media root and
# you have to change alias in corresponding nginx location also (/videos/)
'VIDEO_ARCHIVE_UPLOAD_PATH': '/var/lib/findface-security/uploads/videos/',

# additional events features.
# make sure that corresponding extractors
# are licensed and enabled at findface-extraction-api config file.
# available features are: gender, age, emotions, beard, glasses, medmask.
'EVENTS_FEATURES': [],

# feature specific confidence thresholds
'LIVENESS_THRESHOLD': 0.85,
'EMOTIONS_THRESHOLD': 0.25,
'BEARD_THRESHOLD': 0.7,

# counters full frame saving options:
# `always` - save always
# `detect` - save only if faces or silhouettes have been detected
# `never` - never save full frames
'COUNTERS_SAVE_FULLFRAME': 'always',
'COUNTERS_FULLFRAME_JPEG_QUALITY': 75,
'COUNTERS_THUMBNAIL_JPEG_QUALITY': 75,

# max camera frames_dropped percent
'MAX_CAMERA_DROPPED_FRAMES': {'yellow': 0.1, 'red': 0.3},
# max camera faces_failed percent
'MAX_CAMERA_FAILED_FACES': {'yellow': 0.1, 'red': 0.3},

# -- 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': 0.745,

# 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,

# maximum concurrent video manager jobs for video archives processing
'MAX_VIDEO_ARCHIVE_JOBS': 3,

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

(continues on next page)

(продолжение с предыдущей страницы)

```
'REPORT_FULLFRAME_JPEG_QUALITY': 75,
'REPORT_FULLFRAME_MAX_HEIGHT': 250,

# -- Startup tests --

# required services availability test
'SERVICES_AVAILABILITY_TEST': True,

# enable saving audit logs to PostgreSQL
'ENABLE_AUDIT_LOGS': True,

# -- FFSecurity Onvif --

# auth credentials for ffsecurity_onvif
# ONVIF_CREDENTIALS = [
#     {
#         "hostnames": ["192.168.1.64", "2a00:1370:8117:ab87:a614:37ff:fe49:2683
→"],
#         "login": "admin",
#         "password": "admin123"
#     }
# ],
'ONVIF_CREDENTIALS': {},
# list of all hostnames that will be ignored during Onvif service discovery
# ONVIF_IGNORE_LIST = ["192.168.1.217"],
'ONVIF_IGNORE_LIST': [],

# -- Optional parameters --

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

# Edit CUSTOM_FIELDS->dossier_face section to customize dossier face content.
# Below is an example with every field type possible.

# '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'
```

(continues on next page)

(продолжение с предыдущей страницы)

```

#     },
#     {
#         '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'
#     }
# ]
# },
# 'dossier_face': {
#     '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
#         }
#     ]
# }
# },

# maximum event age in seconds than could be added to an episode.
# 'EPISODE_SEARCH_INTERVAL': 60,
# If none of these events matched, new episode is created.

# maximum episode duration (episode is closed after)
# 'EPISODE_MAX_DURATION': 300,

```

(continues on next page)

(продолжение с предыдущей страницы)

```

# if no new event added to an episode during this timeout, episode will be
↳closed.
# 'EPISODE_EVENT_TIMEOUT': 30,

# maximum created thumbnail width
# 'THUMBNAIL_MAX_WIDTH': 320,

# url of the backend which is used for social network search.
# contact support for additional information.
# 'SOCIAL_BACKEND': None,

# additional social backend headers.
# 'SOCIAL_HEADERS': {},

# unacknowledged events notification interval
# 'UNACKNOWLEDGED_NOTIFY_INTERVAL': 1,

# set to True to run all media requests (photos, attachments) through the
# django application for acl checks.
# enabling this setting slightly increases security but
# has severe negative effects on performance.
# you will also have to mark /uploads/ location as 'internal' in nginx config
#
# 'OVERPROTECT_MEDIA': False,
}

# - FindFace Security authorization configuration dictionary -

FFSECURITY_AUTH_CONFIG = {
# available options: face, password, face_and_password, face_or_password
'AUTH_TYPE': 'face_or_password',
'FACE_AUTH_CONFIDENCE': 0.745,
# 180 days by default
'MAXIMUM_SESSION_LENGTH': 15552000,
# session renew works only with face or face_or_password authorization type
'NEED_SESSION_RENEW': False,
'RENEW_SESSION_INTERVAL': 0,
'MAXIMUM_RENEW_ATTEMPTS': 2,
}

# - FindFace Security user interface configuration dictionary -

FFSECURITY_UI_CONFIG = {
"event": {
"features": {
"f_gender_class": ["male", "female"],
"age": {
"f_age_gte": "",
"f_age_lte": ""
},
"f_emotions_class": ["angry", "disgust", "fear", "happy", "sad",
↳"surprise", "neutral"],

```

(continues on next page)

(продолжение с предыдущей страницы)

```

        "f_glasses_class": ["none", "eye", "sun"],
        "f_beard_class": ["none", "beard"],
        "f_liveness_class": ["real", "fake"],
        "f_medmask_class": ["none", "incorrect", "correct"],
    }
},

# Adjustable confidence threshold presets for face matching.
# Please consult with our support before changing.
"confidence_display": [
    {"confidence": 0.00, "color": "#000000", "label": {"ru": "Очень низкий", "en": "Very Low"}},
    {"confidence": 0.65, "color": "#FF0300", "label": {"ru": "Низкий", "en": "Low"}},
    {"confidence": 0.70, "color": "#FFB700", "label": {"ru": "Пониженный", "en": "Slightly Low"}},
    {"confidence": 0.72, "color": "#B8FA00", "label": {"ru": "Нормальный", "en": "Moderate"}},
    {"confidence": 0.75, "color": "#7EFF2B", "label": {"ru": "Повышенный", "en": "Slightly High"}},
    {"confidence": 0.80, "color": "#4DFF60", "label": {"ru": "Высокий", "en": "High"}},
    {"confidence": 0.85, "color": "#1DFF96", "label": {"ru": "Очень высокий", "en": "Very High"}},
]
}

# -- ASGI-server configuration --
# consult support before changing these settings.

# per worker thread pool size.
ASGI_THREADS = 32

UVICORN_SETTINGS = {
    # worker processes count, 'auto' sets it to logical cpu count
    'workers': 'auto',
    'host': 'localhost',
    'port': 8002,
    # websocket worker processes count,
    # 'auto' sets it to logical cpu count, but not more than 8.
    'ws-workers': 'auto',
    'ws-host': 'localhost',
    'ws-port': 8003,
}

# 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

```

(continues on next page)

(продолжение с предыдущей страницы)

```
        "video_archive_events_manager": True,
        "persons": False,
    }
}

# -- Other settings --

# The number of threads in the night clusterization.
# Not recommended values are greater than the number of cores in the processor.
# Consult with support before changing this value.
NUMPY_OMP_NUM_THREADS = 'auto'

# =====
# FINDFACE SECURITY PLUGINS
# =====
# Uncomment lines below to enable plugins. Please consult documentation for
# a plugin specific settings.

# ===== Axxon =====
# INSTALLED_APPS.append('ffsecurity_axxon')

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

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

# ===== Genetec =====
# INSTALLED_APPS.append('ffsecurity_genetec')

# ===== Sova =====
# INSTALLED_APPS.append('ffsecurity_sova')

# ===== Sigur =====
# keep in mind, that SIGUR plugin also uses CUSTOM_FIELDS and THUMBNAI_MAX_WIDTH
# settings
# INSTALLED_APPS.append('ffsecurity_sigur')
# SIGUR = {
#     'LOGIN': 'admin',
#     'PASSWORD': 'admin',
```

(continues on next page)

(продолжение с предыдущей страницы)

```
# 'MF_SELECTOR': 'biggest', # what to do with several faces in sigur person
↳photo; allowed ['biggest', 'reject']
# 'ONLY_RT_EVENTS': True, # only events with bs_type == realtime,
# 'EVENT_DELAY': 0.004 # minimum time between two events of same person in
↳seconds. If interval between two events with same person is less, than this value,
↳ second event will be dropped
# }

# ===== CryptoPRO authentication =====
# INSTALLED_APPS.append('ffsecurity_cproauth')
# REST_FRAMEWORK['DEFAULT_AUTHENTICATION_CLASSES'] = [
#     'ffsecurity.auth.TokenAuthentication',
#     'ffsecurity_cproauth.auth.CryptoProOrTokenAuthentication'
# ]

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

# token must be identical on master and slave
# use pugen -s 64 1
# SYNC_TOKEN = 'change_me'
# rrule that defines sync schedule
# SYNC_SCHEDULE = 'RRULE:FREQ=DAILY;WKST=MO;BYHOUR=4;BYMINUTE=0'
# if True synchronization will occur on FindFace Security startup and restart
# SYNC_AT_STARTUP = False
# if True synchronization will occur immediately after creating synchronization for
↳dossier list
# SYNC_AT_CREATION = False

# ===== Puppeteer =====
# INSTALLED_APPS.append('ffsecurity_puppeteer')

# PUPPETEER_CONFIG = {
#     'UNSAVED_RESULTS_DELETION_TIMEOUT': 3600, # maximum lifetime of search
↳results not saved involuntarily
#     'REMOTE_MONITORING_SYNC_INTERVAL': 600, # monitoring data
↳synchronization interval, seconds
#     'ENABLE_DAILY_SEARCH': True, # daily search activation
↳(default False)
#     'DAILY_SEARCH_PUSH_HOUR': 2, # daily search dossiers
↳synchronization hour
#     'DAILY_SEARCH_PULL_HOUR': 6, # hour in which results of
↳daily search will be obtained
#     'puppets': [
#         {
#             'id': 'first_puppet', # puppet ID
#             'url': 'http://1.1.1.1:8010/', # puppet URL
#             'token': 'first_puppet_token', # use pugen -s 64 1 (should
↳match the token in puppet)
```

(continues on next page)

(продолжение с предыдущей страницы)

```
#           'facen_model': 'jackfruit_480'           # face model in puppet
#       },
#       {
#           'id': 'second_puppet',
#           'url': 'http://1.1.1.1:8010/',
#           'token': 'second_puppet_token',
#
#           # if remote installation has a different face model than the one
#           →used in FFSecurity -
#           # you need to specify its name and ExtractionAPI URL where the
#           →corresponding face model is specified
#           'facen_model': 'grapefruit_480',
#           'extractor': 'http://127.0.0.1:18667',
#       },
#   ]
# }
#
# ===== Vns =====
# A plugin for using FindFace Security as a puppeteer server
# INSTALLED_APPS.append('ffsecurity_vns')
#
# VNS_CONFIG = {
#     'USERS': {
#         'user1': 'token1',
#         'user2': 'token2'
#     },
#     'MONITORING_THRESHOLD': 0.75,
#     'DAILY': {
#         'ENABLED': False,
#         'THRESHOLD': 0.75,
#         'START_TIME': "00:00:00"
#     }
# }
# }
```

- 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 "' " ". "
```

- Запустите сервисы.

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

- 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
```

Важно: Для того чтобы сохранить совместимость ПК FindFace 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
```

1.5.3 Дополнительное развертывание findface-video-worker на удаленных серверах

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

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

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

1. На сервере ПК FindFace откройте файл конфигурации `/etc/findface-security/config.py` и убедитесь, что параметр `ROUTER_URL` содержит внешний IP-адрес сервера ПК FindFace, а не `localhost`. Экземпляры `findface-video-worker` на удаленных серверах будут использовать данный адрес для отправки лиц.

```
sudo vi /etc/findface-security/config.py
...
'ROUTER_URL': 'http://192.168.0.12',
...
```

2. Загрузите файл инсталлятора `findface-security-and-server-4.5.run`.
3. Поместите файл `.run` в любой каталог на сервере установки (например, `/home/username`).
4. Из данного каталога сделайте файл `.run` исполняемым.

```
chmod +x findface-security-and-server-4.5.run
```

5. Запустите файл `.run`.

```
sudo ./findface-security-and-server-4.5.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 ПК FindFace.

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-4.5.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 ПК FindFace 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
```

1.5.4 Установка моделей нейронных сетей

Для обнаружения и идентификации лиц и их атрибутов (пол, возраст, эмоции, борода, очки, медицинская маска и т. д.) `findface-extraction-api` использует нейронные сети.

Если необходим ручной запуск установки моделей, выполните следующие действия:

1. Запустите подготовленный файл `findface-security-and-server-4.5.run`.

```
sudo ./findface-security-and-server-4.5.run
```

2. Тип установки: `Fully customized installation`.
3. Выберите устанавливаемый компонент ПК FindFace: `findface-data`. Для этого сначала снимите выделение со всех компонентов, введя в командной строке `-*`, затем введите порядковый номер компонента: 1. Введите `done` для сохранения выбора и перехода к следующему шагу.
4. Выберите модели для установки. После этого процесс установки будет автоматически запущен.

Примечание: Вы можете найти установленные модели для распознавания лиц и атрибутов лиц в каталогах `/usr/share/findface-data/models/face/` и `/usr/share/findface-data/models/faceattr/` соответственно.

```
ls /usr/share/findface-data/models/face/
ifruit_320.cpu.fnk  ifruit_320.gpu.fnk  jackfruit_160.cpu.fnk  jackfruit_160.gpu.fnk
↪ jackfruit_320.cpu.fnk  jackfruit_320.gpu.fnk  jackfruit_480.cpu.fnk  jackfruit_480.gpu.
↪ fnk

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

1.5.5 Полностью настраиваемая установка

Консольный инсталлятор ПК FindFace предоставляет несколько вариантов установки, в том числе полностью настраиваемый вариант (установку отдельно выбранных пакетов). Данный вариант в основном используется при развертывании ПК FindFace в сильно распределенной среде.

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

- Устанавливаемый продукт: ПК FindFace.
- Тип установки: `Fully customized installation`.
- Устанавливаемые компоненты ПК FindFace: 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` (для выбора `findface-data` и `findface-extraction-api`), `13` (`findface-tarantool-server`) или `9` (`findface-upload`). Введите `done` для сохранения выбора и перехода к следующему шагу.
- Связанные вопросы, такие как как тип ускорения: CPU или GPU.

1.5.6 Типичная установка в кластере

Данный раздел посвящен развертыванию ПК FindFace в кластерной среде.

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

Развертывание ПК FindFace в кластере может быть необходимо по следующим причинам:

- Нужно распределить высокую нагрузку 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`

- Нужно распределить высокую нагрузку при извлечении биометрических образцов.
- В поиске задействовано большое количество лиц, что требует реализации распределенной базы данных.

Перед тем как приступить к развертыванию, постройте архитектурную схему с учетом будущей нагрузки системы и выделенных под нее аппаратных ресурсов (см. *Системные 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 биометрических образцов 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 развертывания 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*.

Развертывание FindFace в кластерной среде состоит из следующих этапов:

- *Развертывание центрального сервера*
 - *Развертывание серверов обработки видео*
 - *Развертывание биометрических серверов*
 - *Распределение нагрузки между биометрическими серверами*

- Организация распределенной базы данных
- Настройка сетевого взаимодействия

Развертывание центрального сервера

Для развертывания центрального сервера ПК FindFace выполните следующие действия:

1. На выделенном физическом сервере *установите* ПК FindFace из инсталлятора следующим образом:
 - Устанавливаемый продукт: ПК FindFace.
 - Тип установки: **Single server, multiple video workers**. В этом случае ПК FindFace 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 ПК FindFace:

```
#####
#                               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**.

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

## Address to accept incoming client connections (IP:PORT)
## type:string env:CFG_LISTEN longopt:--listen
listen = 0.0.0.1:3133
```

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 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-security-and-server-4.5.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 биометрических серверов

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

- Устанавливаемый продукт: ПК FindFace.
- Тип установки: Fully customized installation.
- Устанавливаемые компоненты ПК FindFace: `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`.
- Устанавливаемые модели нейронных сетей: CPU/GPU-модель для извлечения биометрических данных лица (обязательна для установки) и (опционально) CPU/GPU модели для распознавания пола, возраста, эмоций, очков, бороды и медицинской маски. Для того чтобы выбрать нужные модели, сначала снимите установленное по умолчанию выделение всех моделей, введя в командной строке `-*`, затем введите порядковые номера нужных моделей через пробел, например: `8 2` для выбора соответственно GPU-модели для извлечения биометрических данных и GPU-модели для распознавания возраста. Введите `done` для сохранения выбора и перехода к следующему шагу. Удостоверьтесь, что для каждой модели выбран 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 CPU-, так и GPU-модели. Подробнее см. *Распознавание атрибутов лица*.

Доступны следующие модели:

Атрибут лица	Ускорение	Пакет
биометрия лица	CPU	findface-data-ifruit-320-cpu_3.0.0_all.deb findface-data-jackfruit-160-cpu_3.0.0_all.deb findface-data-jackfruit-320-cpu_3.0.0_all.deb findface-data-jackfruit-480-cpu_3.0.0_all.deb
	GPU	findface-data-ifruit-320-gpu_3.0.0_all.deb findface-data-jackfruit-160-gpu_3.0.0_all.deb findface-data-jackfruit-320-gpu_3.0.0_all.deb findface-data-jackfruit-480-gpu_3.0.0_all.deb
возраст	CPU	findface-data-age.v1-cpu_3.0.0_all.deb
	GPU	findface-data-age.v1-gpu_3.0.0_all.deb
пол	CPU	findface-data-gender.v2-cpu_3.0.0_all.deb
	GPU	findface-data-gender.v2-gpu_3.0.0_all.deb
эмоции	CPU	findface-data-emotions.v1-cpu_3.0.0_all.deb
	GPU	findface-data-emotions.v1-gpu_3.0.0_all.deb
очки	CPU	findface-data-glasses3.v0-cpu_3.0.0_all.deb
	GPU	findface-data-glasses3.v0-gpu_3.0.0_all.deb
борода	CPU	findface-data-beard.v0-cpu_3.0.0_all.deb
	GPU	findface-data-beard.v0-gpu_3.0.0_all.deb
медицинская маска	CPU	findface-data-medmask3.v2-cpu_3.0.0_all.deb
	GPU	findface-data-medmask3.v2-gpu_3.0.0_all.deb

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-security-and-server-4.5.run -f /tmp/<findface-installer-*>.json
```

После развертывания биометрических серверов *распределите* между ними нагрузку.

Распределение нагрузки между биометрическими серверами

Распределение нагрузки между несколькими биометрическими серверами 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 ПК FindFace (172.168.1.9), 2 других на дополнительных удаленных серверах (172.168.1.10, 172.168.1.11). Если в системе присутствует большее количество биометрических серверов, 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 биометрических серверов. Им может стать центральный сервер ПК FindFace (рекомендуется) или любой другой сервер с установленным `nginx`.

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

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 биометрических серверов. В директиве `server` в параметре `listen` укажите номер слушающего порта сервера шлюза. Вам потребуется указать данный порт при настройке *распределенной сети* ПК FindFace.

```
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 биометрических серверов 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 биометрического сервера 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 биометрическом сервере.

```
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* ПК FindFace.

Организация распределенной базы данных

Компонент `findface-tarantool-server` соединяет базу данных 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 лиц. При развертывании `findface-tarantool-server` из инсталлятора шарды создаются автоматически с учетом аппаратной конфигурации физического сервера.

Для того чтобы распределить биометрическую базу данных, установите `findface-tarantool-server` на каждом сервере базы данных. Ответьте на вопросы инсталлятора следующим образом:

- Устанавливаемый продукт: ПК FindFace.
- Тип установки: Fully customized installation.
- Устанавливаемые компоненты ПК FindFace: `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: 13`. 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-security-and-server-4.5.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* ПК FindFace. 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 ПК FindFace 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 биометрических серверов 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 лиц.

```
...
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 ПК FindFace, 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>	(Адрес) IP-адрес или URL, который используется для доступа к веб-интерфейсу ПК FindFace. Если параметр не задан, система автоматически определяет его как внешний IP-адрес. Для доступа в ПК FindFace 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 лиц.
<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 лица от экземпляра(ов) <code>findface-video-worker</code> .
<code>SF_API_ADDRESS</code>	Адрес сервера <code>findface-sf-api</code> .
<code>EXTRACT_IP_ADDR</code>	IP-адрес и слушающий порт сервера, являющегося <i>шлюзом</i> для биометрических серверов с настроенной балансировкой нагрузки.

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

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

(continues on next page)

(продолжение с предыдущей страницы)

```
EXTERNAL_ADDRESS = 'http://127.0.0.1'

...
FFSECURITY = {
  'VIDEO_DETECTOR_TOKEN': '7ce2679adfc4d74edcf508bea4d67208',
  ...
  'EXTRACTION_API': 'http://172.168.1.9:18667/',
  '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 ПК FindFace в кластерной среде будет завершена.

Важно: Для того чтобы сохранить совместимость ПК FindFace 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
```

1.5.7 Добавление репозитория NVIDIA и установка драйверов (только GPU)

ПК FindFace на GPU требует предварительной установки драйверов NVIDIA.

Для добавления репозитория NVIDIA и установки драйверов выполните следующие действия:

1. Загрузите файл инсталлятора `findface-security-and-server-4.5.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-4.5.run
```

4. Запустите файл `.run`.

```
sudo ./findface-security-and-server-4.5.run
```

5. Выберите продукт для установки: `NVIDIA CUDA drivers`.

1.6 Обслуживание и устранение неисправностей

1.6.1 Резервное копирование и восстановление хранилищ данных

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

- Биометрическая база данных на основе Tarantool, в которой хранятся биометрические образцы (векторы признаков) и события идентификации лиц.
- 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 лиц и нормализованные изображения лиц.
- Каталог `/var/lib/ffupload/`, в котором хранятся только такие артефакты событий, как миниатюры лиц.

В этом разделе:

- *Резервное копирование и восстановление биометрической базы данных*
 - *Утилиты*
 - *Резервное копирование базы данных*
 - *Восстановление базы данных*
- *Резервное копирование и восстановление основной базы данных*
- *Резервное копирование и восстановление артефактов*

Резервное копирование и восстановление биометрической базы данных

В биометрической базе данных на основе Tarantool есть 3 галереи:

- `ffsec_dossier_face`: биометрические образцы, извлеченные из фотографий в досье.
- `ffsec_events`: биометрические образцы, извлеченные из лиц, обнаруженных на видео.
- `ffsec_persons`: центроиды персон (виртуальные биометрические образцы, усредненные по всем лицам персоны) и метаданные.

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

Для предотвращения потери данных создание резервной копии биометрической базы данных рекомендуется выполнять по крайней мере 1 раз в неделю. В целом частота резервного копирования зависит от количества досье и событий распознавания лиц, а также от доступного дискового пространства.

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

Утилиты

Для резервного копирования и восстановления биометрической базы данных ПК FindFace 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 биометрической базы данных 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_dossier_face.json`, `ffsec_events.json` и `ffsec_persons.json`. Эти файлы содержат все данные, необходимые для полного восстановления галерей.

Восстановление базы данных

Для восстановления биометрической базы данных запустите утилиту `findface-storage-api-restore` для всех файлов в папке 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 ПК FindFace 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 ПК FindFace, такие как загруженные фотографии, видеофайлы и артефакты событий (полные кадры, миниатюры лиц и нормализованные изображения лиц) 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
```

1.6.2 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 биометрической базы данных 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 биометрические образцы.

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
```

- 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
```

- 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>

- 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/
```

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

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

(continues on next page)

(продолжение с предыдущей страницы)

```
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 (`jackfruit_480.cpu.fnk` в примере).

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

face: face/ifruit_320.cpu.fnk -> face: face/jackfruit_480.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:
```

(continues on next page)

(продолжение с предыдущей страницы)

```

- master: http://127.0.0.1:8111/v2/
  slave: ""
- master: http://127.0.0.1:8112/v2/
  slave: ""
- master: http://127.0.0.1:8113/v2/
  slave: ""
- master: http://127.0.0.1:8114/v2/
  slave: ""
workers_num: 3
faces_limit: 100
extraction_batch_size: 8
normalized_storage:
  type: webdav
  enabled: True
  webdav:
    upload-url: http://127.0.0.1:3333/uploads/
  s3:
    endpoint: ""
    bucket-name: ""
    access-key: ""
    secret-access-key: ""
    secure: False
    region: ""
    public-url: ""
    operation-timeout: 30

```

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

```

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

```

(continues on next page)

(продолжение с предыдущей страницы)

```
- 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 биометрической базы данных.

11. Перенесите схему базы данных из ПК FindFace 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
```

1.6.3 Изменение структуры биометрической базы данных

В некоторых случаях вам может потребоваться применить новую структурную схему к биометрической базе данных Tarantool, например, при обновлении до последней версии продукта или если необходимо усовершенствовать структуру базы данных, добавив в нее дополнительные параметры, расширенные метаданные лиц и т. д.

В этом разделе:

- *О структуре биометрической базы данных*
- *Изменение структуры*

О структуре биометрической базы данных

В ПК FindFace 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 лица;
- `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` приведен ниже:

```
meta_scheme = {
  -- internal.normalized_id:
  {
    default = '',
    field_type = 'string',
    id = 1,
    name = 'normalized_id',
  },
  -- internal.feat:
  {
    default = '',
    field_type = 'string',
    id = 2,
    name = 'feat',
  },
}
```

(continues on next page)

(продолжение с предыдущей страницы)

```

-- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:acknowledged:
{
  default = 0,
  field_type = 'unsigned',
  id = 3,
  name = 'm:acknowledged',
},
-- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:acknowledged_by:
{
  default = 0,
  field_type = 'unsigned',
  id = 4,
  name = 'm:acknowledged_by',
},
-- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:acknowledged_date:
{
  default = 0,
  field_type = 'unsigned',
  id = 5,
  name = 'm:acknowledged_date',
},
-- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:acknowledged_reaction:
{
  default = '',
  field_type = 'string',
  id = 6,
  name = 'm:acknowledged_reaction',
},
-- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:camera:
{
  default = 0,
  field_type = 'unsigned',
  id = 7,
  name = 'm:camera',
},
-- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:camera_group:
{
  default = 0,
  field_type = 'unsigned',
  id = 8,
  name = 'm:camera_group',
},
-- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:confidence:
{
  default = "10000000000000000000",
  field_type = 'unsigned',
  id = 9,
  name = 'm:confidence',
},
-- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:created_date:
{
  default = 0,

```

(continues on next page)

(продолжение с предыдущей страницы)

```

    field_type = 'unsigned',
    id = 10,
    name = 'm:created_date',
},
-- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:episode:
{
    default = 0,
    field_type = 'unsigned',
    id = 11,
    name = 'm:episode',
},
-- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:episode_open:
{
    default = 0,
    field_type = 'unsigned',
    id = 12,
    name = 'm:episode_open',
},
-- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:f_age:
{
    default = "10000000000000000000",
    field_type = 'unsigned',
    id = 13,
    name = 'm:f_age',
},
-- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:f_beard_class:
{
    default = '',
    field_type = 'string',
    id = 14,
    name = 'm:f_beard_class',
},
-- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:f_beard_score:
{
    default = "10000000000000000000",
    field_type = 'unsigned',
    id = 15,
    name = 'm:f_beard_score',
},
-- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:f_emotions_class:
{
    default = '',
    field_type = 'string',
    id = 16,
    name = 'm:f_emotions_class',
},
-- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:f_emotions_score:
{
    default = "10000000000000000000",
    field_type = 'unsigned',
    id = 17,
    name = 'm:f_emotions_score',
}

```

(continues on next page)

(продолжение с предыдущей страницы)

```

},
-- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:f_gender_class:
{
    default = '',
    field_type = 'string',
    id = 18,
    name = 'm:f_gender_class',
},
-- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:f_gender_score:
{
    default = "10000000000000000000",
    field_type = 'unsigned',
    id = 19,
    name = 'm:f_gender_score',
},
-- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:f_glasses_class:
{
    default = '',
    field_type = 'string',
    id = 20,
    name = 'm:f_glasses_class',
},
-- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:f_glasses_score:
{
    default = "10000000000000000000",
    field_type = 'unsigned',
    id = 21,
    name = 'm:f_glasses_score',
},
-- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:f_liveness_class:
{
    default = '',
    field_type = 'string',
    id = 22,
    name = 'm:f_liveness_class',
},
-- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:f_liveness_score:
{
    default = "10000000000000000000",
    field_type = 'unsigned',
    id = 23,
    name = 'm:f_liveness_score',
},
-- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:f_medmask_class:
{
    default = '',
    field_type = 'string',
    id = 24,
    name = 'm:f_medmask_class',
},
-- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:f_medmask_score:
{

```

(continues on next page)

(продолжение с предыдущей страницы)

```

    default = "10000000000000000000",
    field_type = 'unsigned',
    id = 25,
    name = 'm:f_medmask_score',
},
-- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:thumbnail:
{
    default = '',
    field_type = 'string',
    id = 28,
    name = 'm:thumbnail',
},
-- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:frame:
{
    default = '',
    field_type = 'string',
    id = 29,
    name = 'm:frame',
},
-- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:frame_coords_bottom:
{
    default = 0,
    field_type = 'unsigned',
    id = 30,
    name = 'm:frame_coords_bottom',
},
-- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:frame_coords_left:
{
    default = 0,
    field_type = 'unsigned',
    id = 31,
    name = 'm:frame_coords_left',
},
-- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:frame_coords_right:
{
    default = 0,
    field_type = 'unsigned',
    id = 32,
    name = 'm:frame_coords_right',
},
-- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:frame_coords_top:
{
    default = 0,
    field_type = 'unsigned',
    id = 33,
    name = 'm:frame_coords_top',
},
-- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:pk:
{
    default = 0,
    field_type = 'unsigned',
    id = 34,

```

(continues on next page)

(продолжение с предыдущей страницы)

```

    name = 'm:pk',
},
-- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:matched:
{
    default = 0,
    field_type = 'unsigned',
    id = 35,
    name = 'm:matched',
},
-- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:matched_dossier:
{
    default = 0,
    field_type = 'unsigned',
    id = 36,
    name = 'm:matched_dossier',
},
-- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:matched_face:
{
    default = 0,
    field_type = 'unsigned',
    id = 37,
    name = 'm:matched_face',
},
-- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:matched_lists:
{
    default = {},
    field_type = 'set[unsigned]',
    id = 38,
    name = 'm:matched_lists',
},
-- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:normalized_photo:
{
    default = '',
    field_type = 'string',
    id = 39,
    name = 'm:normalized_photo',
},
-- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:quality:
{
    default = "10000000000000000000",
    field_type = 'unsigned',
    id = 40,
    name = 'm:quality',
},
-- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:scores:
{
    default = '',
    field_type = 'string',
    id = 41,
    name = 'm:scores',
},
-- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:temperature:

```

(continues on next page)

(продолжение с предыдущей страницы)

```

{
  default = "0",
  field_type = 'unsigned',
  id = 42,
  name = 'm:temperature',
},
-- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:video_source:
{
  default = 0,
  field_type = 'unsigned',
  id = 43,
  name = 'm:video_source',
},
-- <class 'ffsecurity.entities_tnt.dossier_face.models.DossierFace'>.m:dossier:
{
  default = 0,
  field_type = 'unsigned',
  id = 44,
  name = 'm:dossier',
},
-- <class 'ffsecurity.entities_tnt.dossier_face.models.DossierFace'>.m:modified_date:
{
  default = 0,
  field_type = 'unsigned',
  id = 45,
  name = 'm:modified_date',
},
-- <class 'ffsecurity.entities_tnt.dossier_face.models.DossierFace'>.m:source_photo:
{
  default = '',
  field_type = 'string',
  id = 46,
  name = 'm:source_photo',
},
-- <class 'ffsecurity.entities_tnt.dossier_face.models.DossierFace'>.m:source_photo_
↔name:
{
  default = '',
  field_type = 'string',
  id = 47,
  name = 'm:source_photo_name',
},
}
-- Fields referenced by multiple models: m:thumbnail, m:frame_coords_left, m:frame_
↔coords_right, m:pk, m:frame_coords_bottom, m:created_date, m:frame_coords_top
meta_indexes = {'m:episode', 'm:episode_open', 'm:video_source'}

```

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 биометрической базы данных 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
```

- Запустите сервис `findface-security`.

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

См.также:

Пользовательские метаданные лиц досье в Tarantool

1.6.4 Удаление экземпляра продукта

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

Выполните следующие действия:

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

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

- Запустите скрипт.

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

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

1.6.5 Проверка статуса компонентов

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

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

1.6.6 Логи сервисов

При разборе нештатных ситуаций используйте логи сервисов ПК FindFace, содержащие подробную детализировку всех событий, произошедших в системе.

Для того посмотреть логи для каждого компонента, используя команду `journalctl -u <component>`, например:

```
journalctl -u findface-extraction-api
```

Важно: Для того чтобы включить хранение логов `journald` на жестком диске в ОС Ubuntu, в файле `/etc/systemd/journald.conf` раскомментируйте и измените параметр `Storage`:

```
sudo vi /etc/systemd/journald.conf
...
[Journal]
Storage=persistent
```

При необходимости также раскомментируйте и измените значение параметра `SystemMaxUse`. Данный параметр определяет в процентах максимальный объем логов на жестком диске (по умолчанию 10%).

```
SystemMaxUse=15
```

См. также:

Журнал действий пользователей

1.6.7 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* ПК FindFace и статусе `findface-ntls`, выполните в консоли сервера `findface-ntls` следующую команду:

```
curl http://localhost:3185/license.json -s | jq
```

Ответ будет возвращен в формате JSON. Одним из наиболее значимых параметров в ответе является `last_updated`. Он показывает в секундах, как давно в последний раз проверялась локальная лицензия.

Интерпретируйте значение параметра `last_updated` следующим образом:

- [0, 5] — все работает отлично.
- (5, 30] — возможно имеют место быть какие-то проблемы со связью, либо с локальным накопителем, где хранятся файлы лицензий.
- (30; 120] — почти наверняка случилось что-то нехорошее.
- (120; ∞) — не удастся получить ответ от источника лицензирования в течение длительного времени. Необходимо вмешательство.
- "valid": false: связь с источником лицензирования так и не была установлена.

```
curl http://localhost:3185/license.json -s | jq
{
  "name": "NTLS",
  "time": 1565186356,
  "type": "online",
  "license_id": "61063ce4b86945e1b70c3bdbedea453b",
  "generated": 1514467939,
  "last_updated": 5,
  "valid": {
    "value": true,
    "description": ""
  },
  "source": "/opt/ntech/license/import_
↪b68d7b7ec9a7310d18832035318cff0c9ddf11e3a9ab0ae962fbe48645e196d1.lic",
  "limits": [
    {
      "type": "time",
      "name": "end",
      "value": 1609161621
    },
    {
      "type": "number",
      "name": "faces",
      "value": 9007199254740991,
      "current": 0
    },
    {
      "type": "number",
      "name": "cameras",
      "value": 4294967295,
      "current": 0
    },
    {
      "type": "number",
      "name": "extraction_api",
      "value": 256,
      "current": 0
    },
    {
      "type": "boolean",
      "name": "gender",
      "value": true
    }
  ],
}
```

(continues on next page)

(продолжение с предыдущей страницы)

```
{
  "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": "countries",
  "value": false
},
{
  "type": "boolean",
  "name": "beard",
  "value": false
},
{
  "type": "boolean",
  "name": "glasses",
  "value": false
},
{
  "type": "boolean",
  "name": "liveness",
  "value": false
}
],
"services": [
  {
    "name": "video-worker",
    "ip": "127.0.0.1:53276"
  },
  {
    "name": "FindFace-tarantool",
    "ip": "127.0.0.1:53284"
  },
  {
    "name": "FindFace-tarantool",
```

(continues on next page)

(продолжение с предыдущей страницы)

```
    "ip": "127.0.0.1:53288"  
  }  
]  
}
```

1.6.8 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-*
```

1.6.9 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 ПК FindFace используйте утилиту `cleanup`. Вы можете выборочно удалить следующие данные:

- события, для которых были найдены совпадения в базе данных, и связанные эпизоды,
- события без совпадений и связанные эпизоды,
- полные кадры событий с совпадениями,
- полные кадры событий без совпадений,
- записи счетчика,
- события создания персоны.

Справка по утилите `cleanup` вызывается следующей командой:

```

sudo findface-security cleanup --help
usage: findface-security cleanup [-h] [--as-configured]
                                [--events-matched-age EVENTS_MATCHED_AGE]
                                [--events-unmatched-age EVENTS_UNMATCHED_AGE]
                                [--events-fullframe-matched-age EVENTS_FULLFRAME_
↳MATCHED_AGE]
                                [--events-fullframe-unmatched-age EVENTS_FULLFRAME_
↳UNMATCHED_AGE]
                                [--counter-records-age COUNTER_RECORDS_AGE]
                                [--person-events-age PERSON_EVENTS_AGE]
                                [--configuration CONFIGURATION] [--version]
                                [-v {0,1,2,3}] [--settings SETTINGS]
                                [--pythonpath PYTHONPATH] [--traceback]
                                [--no-color]

Delete FFSecurity entities
optional arguments:
  -h, --help            show this help message and exit
  --as-configured       Apply config age options for events, counter records
                        and persons. Can't be used with other arguments.
  --events-matched-age EVENTS_MATCHED_AGE
                        Minimum age in days of matched events to clean up
  --events-unmatched-age EVENTS_UNMATCHED_AGE
                        Minimum age in days of unmatched events to clean up
  --events-fullframe-matched-age EVENTS_FULLFRAME_MATCHED_AGE
                        Minimum age in days of matched events fullframes to
                        clean up
  --events-fullframe-unmatched-age EVENTS_FULLFRAME_UNMATCHED_AGE
                        Minimum age in days of unmatched 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
  --configuration CONFIGURATION
                        The name of the configuration class to load, e.g.
                        "Development". If this isn't provided, the
                        DJANGO_CONFIGURATION environment variable will be
                        used.
  --version            show program's version number and exit
  -v {0,1,2,3}, --verbosity {0,1,2,3}
                        Verbosity level; 0=minimal output, 1=normal output,
                        2=verbose output, 3=very verbose output
  --settings SETTINGS
                        The Python path to a settings module, e.g.
                        "myproject.settings.main". If this isn't provided, the
                        DJANGO_SETTINGS_MODULE environment variable will be
                        used.
  --pythonpath PYTHONPATH
                        A directory to add to the Python path, e.g.
                        "/home/djangoprojects/myproject".
  --traceback          Raise on CommandError exceptions
  --no-color           Don't colorize the command output.

```

Для полного удаления событий и эпизодов старше определенного количества дней используйте ар-

гументы `-events-matched-age/-events-unmatched-age`. Например, для удаления событий и эпизодов без совпадений старше 5 дней выполните команду:

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

Для удаления событий с совпадениями старше 5 дней выполните команду:

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

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

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

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

1.6.10 Отключение сервисов

Существует возможность отключить следующие сервисы ПК FindFace, 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`.

```
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,
    }
}
```

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.

1.6.11 Сброс пароля

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

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

1.6.12 Перенос данных на другой диск

Высокая нагрузка на диск может приводить к задержкам при поступлении событий. В тяжелых случаях возможна полная неработоспособность ПК FindFace. Одним из способов снижения нагрузки на диск является перенос хранилищ данных ПК FindFace 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 ПК FindFace 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
```

1.7 Приложения

1.7.1 Подробно о компонентах

findface-extraction-api

Компонент `findface-extraction-api` с помощью нейронных сетей обнаруживает лицо на изображении, извлекает из лица биометрический образец, а также распознает пол, возраст, эмоции и другие атрибуты лица.

Компонент взаимодействует с сервисом `findface-sf-api` следующим образом:

- Получает от него фотографию с лицом или нормализованное изображение лица.
- Возвращает координаты рамки с лицом, а также вектор признаков, данные о поле, возрасте, эмоциях и других атрибутах лица (если они были запрошены `findface-sf-api`).

Полный список функций:

- детекция (обнаружение) лица на исходном изображении с возвращением координат рамки с лицом,
- получение из исходного изображения нормализованного изображения лица,

- извлечение из нормализованного изображения лица вектора признаков (биометрического образца),
- распознавание атрибутов лица (пол, возраст, эмоции, борода, очки, медицинская маска 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`, однако содержимое данного файла на CPU и GPU отличается.

Файл конфигурации сервиса на CPU:

```
detectors:
  max_batch_size: 1
  instances: 1
  models:
    cheetah:
      aliases:
        - face
        - nnd
      model: facedet/cheetah.cpu.fnk
      options:
        min_object_size: 32
        resolutions:
          - 256x256
          - 384x384
          - 512x512
          - 768x768
          - 1024x1024
          - 1536x1536
          - 2048x2048
      quality_estimator: true
normalizers:
  max_batch_size: 8
  instances: 1
  models:
    carlicplate:
      model: ''
    crop1x:
      model: ''
    crop2x:
      model: facenorm/crop2x.v2_maxsize400.cpu.fnk
    cropbbox:
      model: ''
    norm200:
      model: facenorm/bee.v2.cpu.fnk
extractors:
  max_batch_size: 8
  instances: 1
  models:
    age: ''
    beard: ''
    carattr_color: ''
```

(continues on next page)

(продолжение с предыдущей страницы)

```
carattr_description: ''
carattr_license_plate: ''
carattr_make: ''
carattr_trash: ''
countries47: ''
emotions: ''
face: face/jackfruit_480.cpu.fnk
gender: ''
glasses3: ''
headpose: ''
liveness: faceattr/liveness.alleyn.v2.cpu.fnk
luminance_overexposure: ''
luminance_underexposure: ''
medmask3: ''
pedattr_color: ''
pedestrian: ''
quality: faceattr/quality.v1.cpu.fnk
sharpness: ''
validity: ''
ascend_device: 0
gpu_device: 0
models_root: /usr/share/findface-data/models
cache_dir: /var/cache/findface/models_cache
listen: 127.0.0.1:18666
license_ntls_server: 127.0.0.1:3133
fetch:
  enabled: true
  size_limit: 10485760
max_dimension: 6000
allow_cors: false
ticker_interval: 5000
debug: false
prometheus:
  timing_buckets:
    - 0.001
    - 0.005
    - 0.01
    - 0.02
    - 0.03
    - 0.05
    - 0.1
    - 0.2
    - 0.3
    - 0.5
    - 0.75
    - 0.9
    - 1
    - 1.1
    - 1.3
    - 1.5
    - 1.7
    - 2
```

(continues on next page)

(продолжение с предыдущей страницы)

```
- 3
- 5
- 10
- 20
- 30
- 50
resolution_buckets:
- 10000
- 20000
- 40000
- 80000
- 100000
- 200000
- 400000
- 800000
- 1e+06
- 2e+06
- 3e+06
- 4e+06
- 5e+06
- 6e+06
- 8e+06
- 1e+07
- 12000000.0
- 15000000.0
- 18000000.0
- 2e+07
- 3e+07
- 5e+07
- 1e+08
faces_buckets:
- 0
- 1
- 2
- 5
- 10
- 20
- 50
- 75
- 100
- 200
- 300
- 400
- 500
- 600
- 700
- 800
- 900
- 1000
```

Файл конфигурации сервиса на GPU:

```
detectors:
  max_batch_size: 1
  instances: 1
  models:
    cheetah:
      aliases:
        - face
        - nnd
      model: facedet/cheetah.gpu.fnk
      options:
        min_object_size: 32
        resolutions:
          - 256x256
          - 384x384
          - 512x512
          - 768x768
          - 1024x1024
          - 1536x1536
          - 2048x2048
      quality_estimator: true
normalizers:
  max_batch_size: 8
  instances: 1
  models:
    carlicplate:
      model: ''
    crop1x:
      model: ''
    crop2x:
      model: facenorm/crop2x.v2_maxsize400.gpu.fnk
    cropbbox:
      model: ''
    norm200:
      model: facenorm/bee.v2.gpu.fnk
extractors:
  max_batch_size: 8
  instances: 1
  models:
    age: ''
    beard: ''
    carattr_color: ''
    carattr_description: ''
    carattr_license_plate: ''
    carattr_make: ''
    carattr_trash: ''
    countries47: ''
    emotions: ''
    face: face/jackfruit_480.gpu.fnk
    gender: ''
    glasses3: ''
    headpose: ''
```

(continues on next page)

(продолжение с предыдущей страницы)

```

liveness: faceattr/liveness.alleyn.v2.gpu.fnk
luminance_overexposure: ''
luminance_underexposure: ''
medmask3: ''
pedattr_color: ''
pedestrian: ''
quality: faceattr/quality.v1.gpu.fnk
sharpness: ''
validity: ''
ascend_device: 0
gpu_device: 0
models_root: /usr/share/findface-data/models
cache_dir: /var/cache/findface/models_cache
listen: 127.0.0.1:18666
license_ntls_server: 127.0.0.1:3133
fetch:
  enabled: true
  size_limit: 10485760
max_dimension: 6000
allow_cors: false
ticker_interval: 5000
debug: false
prometheus:
  timing_buckets:
    - 0.001
    - 0.005
    - 0.01
    - 0.02
    - 0.03
    - 0.05
    - 0.1
    - 0.2
    - 0.3
    - 0.5
    - 0.75
    - 0.9
    - 1
    - 1.1
    - 1.3
    - 1.5
    - 1.7
    - 2
    - 3
    - 5
    - 10
    - 20
    - 30
    - 50
  resolution_buckets:
    - 10000
    - 20000
    - 40000

```

(continues on next page)

(продолжение с предыдущей страницы)

```

- 80000
- 100000
- 200000
- 400000
- 800000
- 1e+06
- 2e+06
- 3e+06
- 4e+06
- 5e+06
- 6e+06
- 8e+06
- 1e+07
- 12000000.0
- 15000000.0
- 18000000.0
- 2e+07
- 3e+07
- 5e+07
- 1e+08
faces_buckets:
- 0
- 1
- 2
- 5
- 10
- 20
- 50
- 75
- 100
- 200
- 300
- 400
- 500
- 600
- 700
- 800
- 900
- 1000
    
```

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>	IP-адрес и порт сервера лицензирования <code>findface-ntls</code> .

В зависимости от нужд вашего бизнеса, вам также может потребоваться включить модели распозна-

вания атрибутов лица, таких как пол, возраст, эмоции, очки и/или борода. 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.

```
models:
  age: faceattr/age.v1.cpu.fnk
  emotions: faceattr/emotions.v1.cpu.fnk
  face: face/jackfruit_480.cpu.fnk
  gender: faceattr/gender.v2.cpu.fnk
  beard: faceattr/beard.v0.cpu.fnk
  glasses3: faceattr/glasses3.v0.cpu.fnk
  medmask3: faceattr/medmask3.v2.cpu.fnk
```

Доступны следующие модели:

Атрибут лица	Ускорение	Параметр в файле конфигурации
биометрия лица	CPU	face: face/ifruit_320.cpu.fnk face: face/jackfruit_160.cpu.fnk face: face/jackfruit_320.cpu.fnk face: face/jackfruit_480.cpu.fnk
	GPU	face: face/ifruit_320.gpu.fnk face: face/jackfruit_160.gpu.fnk face: face/jackfruit_320.gpu.fnk face: face/jackfruit_480.gpu.fnk
возраст	CPU	age: faceattr/age.v1.cpu.fnk
	GPU	age: faceattr/age.v1.gpu.fnk
пол	CPU	gender: faceattr/gender.v2.cpu.fnk
	GPU	gender: faceattr/gender.v2.gpu.fnk
эмоции	CPU	emotions: faceattr/emotions.v1.cpu.fnk
	GPU	emotions: faceattr/emotions.v1.gpu.fnk
очки	CPU	glasses3: faceattr/glasses3.v0.cpu.fnk
	GPU	glasses3: faceattr/glasses3.v0.gpu.fnk
борода	CPU	beard: faceattr/beard.v0.cpu.fnk
	GPU	beard: faceattr/beard.v0.gpu.fnk
медицинская маска	CPU	medmask3: faceattr/medmask3.v2.cpu.fnk
	GPU	medmask3: faceattr/medmask3.v2.gpu.fnk

Для того чтобы включить модель нейронной сети, предоставляющей *автономный сервис liveness*, укажите ее в параметре `liveness: faceattr/liveness.alley.v2.cpu.fnk/faceattr/liveness.alley.v2.gpu.fnk`.

```
models:
  ...
  liveness: faceattr/liveness.alley.v2.cpu.fnk
  ...

models:
  ...
  liveness: faceattr/liveness.alley.v2.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.

```
models:
  gender: ""
  age: ""
  emotions: ""
```

findface-sf-api

Компонент `findface-sf-api` представляет собой сервис, реализующий HTTP API основного функционала ядра FindFace (обнаружение и распознавание лиц, при этом сами операции обнаружения и распознавания лиц выполняются компонентом `findface-extracton-api`). Взаимодействует с базой биометрических данных Tarantool через компонент `findface-tarantool-server`, а также с компонентами `findface-extraction-api` (обнаружение и распознавание лиц) и `findface-upload` (хранилище исходных изображений и артефактов работы ядра FindFace).

Для обнаружения лица 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 лиц (присутствует в ПК FindFace) компонент `findface-sf-api` получает данные об обнаруженных лицах вместе с правилами их обработки от компонента `findface-facerouter` и затем выполняет полученные директивы (например, сохраняет лица в определенную галерею базы данных).

Примечание: В ПК FindFace, функции `findface-facerouter` выполняет сервис `findface-security`.

Полный список функций:

- реализация HTTP API по части обнаружения и распознавания лиц (операции выполняются `findface-extraction-api`).
- сохранение лиц в базу биометрических данных (через сервис `findface-tarantool-server`),
- сохранение исходных изображений, миниатюр и нормализованных изображений лиц 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`.

```
listen: 127.0.0.1:18411
extraction-api:
  timeouts:
    connect: 5s
    response_header: 30s
    overall: 35s
    idle_connection: 10s
  extraction-api: http://127.0.0.1:18666
storage-api:
  timeouts:
    connect: 5s
    response_header: 30s
    overall: 35s
```

(continues on next page)

(продолжение с предыдущей страницы)

```
idle_connection: 10s
max-idle-conns-per-host: 20
shards:
- master: http://127.0.0.1:8101/v2/
  slave: ''
- master: http://127.0.0.1:8102/v2/
  slave: ''
- master: http://127.0.0.1:8103/v2/
  slave: ''
- master: http://127.0.0.1:8104/v2/
  slave: ''
- master: http://127.0.0.1:8105/v2/
  slave: ''
- master: http://127.0.0.1:8106/v2/
  slave: ''
read_slave_first: false
max_slave_attempts: 2
cooldown: 2s
limits:
url-length: 4096
deny-networks: 127.0.0.0/8,192.168.0.0/16,10.0.0.0/8,::1/128,fe00::/8
body-image-length: 33554432
allow-return-facen: true
cache:
type: memcache
inmemory:
size: 16384
memcache:
nodes:
- 127.0.0.1:11211
timeout: 100ms
dns_cache_timeout: 1m0s
redis:
network: tcp
addr: localhost:6379
password: ''
db: 0
timeout: 5s
normalized-storage:
type: webdav
enabled: true
webdav:
upload-url: http://127.0.0.1:3333/uploads/
timeouts:
connect: 5s
response_header: 30s
overall: 35s
idle_connection: 10s
s3:
endpoint: ''
bucket-name: ''
access-key: ''
```

(continues on next page)

(продолжение с предыдущей страницы)

```
secret-access-key: ''
secure: true
region: ''
public-url: ''
operation-timeout: 30
```

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 лиц.
<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 биометрической базой данных 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 лицах).
- 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 лиц в базу биометрических данных,
- 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`. В кластерной среде файл конфигурации настраивается отдельно для каждого шарда.

```
--
-- Please, read the tarantool documentation at https://www.tarantool.io/en/doc/1.10/
--
box.cfg{
  -- THIS IS NOT HTTP API PORT, it's for admin operations
  listen = '127.0.0.1:32001',

  --Directory to store data
  vinyl_dir = '/opt/nitech/var/lib/tarantool/shard-001',
  work_dir = '/opt/nitech/var/lib/tarantool/shard-001',
  memtx_dir = '/opt/nitech/var/lib/tarantool/shard-001/snapshots',
  wal_dir = '/opt/nitech/var/lib/tarantool/shard-001/xlogs',

  --Maximum mem usage in bytes
  memtx_memory = 200 * 1024 * 1024,

  checkpoint_interval = 3600*4,
  checkpoint_count = 3,

  --uncomment only if you know what you are doing!!! and don't forget box.snapshot()
  -- wal_mode = 'none',

  --if true, tarantool tries to continue if there is an error while reading a snapshot/
  ↪xlog files: skips invalid records, reads as much data as possible and re-builds the ↪
  ↪file
  -- force_recovery = true,
}

pcall(function() box.schema.user.grant('guest', 'execute,read,write', 'universe') end)

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
})

```

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_path</code>	Путь к серверу лицензирования <code>findface-ntls</code> .
<code>memtx_max_size</code>	Максимальный размер оперативной памяти в байтах, который может быть использован шардом Tarantool. Перед изменением данного параметра обратитесь к нашим экспертам за консультацией по адресу support@ntechlab.com .
<code>meta_scheme</code>	Структура базы данных для хранения биометрических параметров. 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 лица, <code>field_type</code> : тип данных, <code>default</code> : значение по умолчанию. Если значение по умолчанию для поля больше <code>'1e14 - 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 биометрической базы данных*.

findface-upload

Компонент `findface-upload` представляет собой веб-сервер на базе WebDAV nginx, который используется для хранения исходных изображений, миниатюр и нормализованных изображений лиц (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 лиц: `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 лиц*
- *Job-задания*

Функции `findface-video-manager`

Сервис `findface-video-manager` является частью модуля видеодетекции лиц и используется для непосредственного управления детекцией лиц 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 лиц.

Полный список функций `findface-video-manager`:

- конфигурирование параметров видеодетектора лиц,
- управление списком видеопотоков для обработки,
- управление видеодетекцией лиц.

Функции `findface-video-worker`

Компонент `findface-video-worker` (или `findface-video-worker-gpu`) является частью модуля видеодетекции лиц и служит для обнаружения лиц «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 лиц вместе с полными кадрами и метаданными, такими как рамка вокруг лица, ID камеры и время детекции, в сервис `findface-facerouter` для дальнейшей обработки.

Примечание: В ПК FindFace функции `findface-facerouter` выполняются сервисом `findface-security`.

Полный список функций `findface-video-manager`:

- обнаружение лиц на видео,
- извлечение нормализованных изображений лиц,
- поиск наилучшего изображения лица,
- дедупликация кадров с лицом (только один кадр на каждое событие распознавания лица).

При обработки видео `findface-video-worker` последовательно использует следующие алгоритмы:

- **Детектор движения.** Данный алгоритм позволяет снизить потребление ресурсов, поскольку детектор лиц включается только по движению в кадре.
- **Детектор лиц.** Алгоритм детектирует, отслеживает и захватывает лица на видео. Может работать одновременно с несколькими лицами в кадре. С помощью встроенной нейронной сети выполняет поиск кадра с лучшим изображением лица. Как только лучшее изображение найдено, отправляет его в компонент `findface-facerouter`.

Подбор лучшего изображения лица 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 лица в компонент `findface-facerouter` на лету после появления лица в поле зрения видеокamеры. Доступны следующие опции отправки (см. `/etc/findface-video-manager.conf` и *Добавление камеры*):

- Если параметр `realtime_post_every_interval: true`, детектор лиц 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 лиц выбирает лучшее изображение лица динамически:
 1. Сначала оценивается качество изображения лица. Если оно превышает некое предустановленное пороговое значение, то лицо отправляется в `findface-facerouter`.
 2. Порог повышается после каждой отправки изображения лица в `findface-facerouter`. Каждый раз, когда детектор лиц получает изображение того же лица лучшего качества, оно отправляется.
 3. При исчезновении лица из поля зрения видеокamеры снова устанавливается пороговое значение по умолчанию.
- Если `realtime_post_first_immediately: true`: детектор лиц отправляет первое лицо из трека сразу после того, как оно пройдет через фильтры качества, размера и региона захвата (ROI), не дожидаясь окончания первого `realtime_post_interval` в режиме реального времени. Способ отправки последующих изображений лица зависит от установленного значения `realtime_post_every_interval`. Если `realtime_post_first_immediately: false`, детектор лиц отправляет первое лицо после окончания первого `realtime_post_interval`.

Буферный режим

Буферный режим требует меньший объем дискового пространства по сравнению с режимом реального времени, поскольку для каждого лица компонент `findface-video-worker` отправляет только одно изображение из трека, но наивысшего качества.

Буферный режим включается через параметр `overall_only` (см. `/etc/findface-video-manager.conf` и *Добавление камеры*).

Настройка видеодетекции лиц

Настройка видеодетектора лиц выполняется через следующие файлы конфигурации:

1. Файл конфигурации компонента `findface-video-manager` `/etc/findface-video-manager.conf`:

```
listen: 127.0.0.1:18810
etcd:
  endpoints: 127.0.0.1:2379
  dial_timeout: 3s
kafka:
  enabled: false
  endpoints: 127.0.0.1:9092
master:
  lease_ttl: 10
  self_url: 127.0.0.1:18811
  self_url_http: 127.0.0.1:18810
rpc:
  listen: 127.0.0.1:18811
  heart_beat_timeout: 4s
router_url: http://127.0.0.1:18820/v0/frame
exp_backoff:
  enabled: false
  min_delay: 1s
  max_delay: 1m0s
  factor: 2
  flush_interval: 2m0s
ntls:
  enabled: false
  url: http://127.0.0.1:3185/
  update_interval: 1m0s
prometheus:
  jobs_processed_duration_buckets:
    - 1
    - 30
    - 60
    - 500
    - 1800
    - 3600
    - 21600
    - .inf
job_scheduler_script: ''
stream_settings:
  ffmpeg_params: []
  md_threshold: 0.002
  md_scale: 0.3
  fd_frame_height: -1
  uc_max_time_diff: 30
  uc_max_dup: 3
  uc_max_avg_shift: 10
  det_period: 8
  realtime: false
  npersons: 4
```

(continues on next page)

(продолжение с предыдущей страницы)

```

disable_drops: false
tracker_threads: 4
parse_sei: false
image_arg: photo
additional_headers: []
additional_body: []
api_timeout: 15000
api_ssl_verify: true
post_uniq: true
min_score: -2
min_d_score: -1000
realtime_dly: 500
realtime_post_perm: false
rot: ''
roi: ''
draw_track: false
send_track: 0
min_face_size: 0
max_face_size: 0
overall: true
only_norm: false
max_candidates: 0
jpeg_quality: 95
ffmpeg_format: ''
stream_settings_gpu:
  play_speed: -1
  filter_min_quality: 0.45
  filter_min_face_size: 1
  filter_max_face_size: 8192
  normalized_only: false
  jpeg_quality: 95
  overall_only: false
  use_stream_timestamp: false
  ffmpeg_params: []
  router_timeout_ms: 15000
  router_verify_ssl: true
  router_headers: []
  router_body: []
  start_stream_timestamp: 0
  imotion_threshold: 0
  rot: ''
  roi: ''
  realtime_post_interval: 1
  realtime_post_every_interval: false
  ffmpeg_format: ''
  disable_drops: false
  router_full_frame_png: false
  router_disable_normalized: false
  crop_fullframe_rot: false
  realtime_post_first_immediately: false
  post_first_track_frame: false
  post_last_track_frame: false

```

(continues on next page)

(продолжение с предыдущей страницы)

```
track_max_duration_frames: 0
send_track_history: false
stream_data_filter: ''
```

Пользовательская настройка `findface-video-manager` выполняется с использованием следующих параметров:

Опция	Описание
<code>etcd -> endpoints</code>	IP-адрес и порт сервиса <code>etcd</code> . Значение по умолчанию: <code>127.0.0.1:2379</code> .
<code>ntls -> enabled</code>	Если <code>true</code> , компонент <code>findface-video-manager</code> отправляет в компонент <code>findface-video-worker</code> задания только на обработку того количества видеокамер, которое указано в лицензии. Значение по умолчанию: <code>false</code> .
<code>ntls -> url</code>	IP-адрес и порт сервера <code>findface-ntls</code> . Значение по умолчанию: <code>http://127.0.0.1:3185/</code> .
<code>router_url</code>	IP-адрес и порт сервера <code>findface-facerouter</code> , который получает обнаруженные лица из <code>findface-video-worker</code> . В ПК FindFace функции <code>findface-facerouter</code> выполняет компонент <code>findface-security</code> . Значение по умолчанию: <code>http://127.0.0.1:18820/v0/frame</code> .

Вы также можете использовать следующие параметры:

Примечание: В разделе файла `stream_settings-gpu` вы найдете настройки, общие для всех видеопотоков. Настройки в данном разделе работают как в GPU, так и CPU-конфигурации. Настройки определенного потока, переданные в `job`-задании, имеют приоритет над настройками в файле конфигурации (см. *Job-задания*).

Примечание: Раздел `stream_settings` файла устарел и необходим только для обратной совместимости.

Опция	Описание
crop_fullframe	Отправлять отправляемые полные кадры по ROT (региону отслеживания). Значение по умолчанию: false.
disable_drop	Включает/отключает отправку в компонент <code>findface-facerouter</code> всех подходящих лиц без пропусков. По умолчанию, если <code>findface-video-worker</code> не обладает достаточными ресурсами для обработки всех кадров с лицами, он отбрасывает некоторые из них. Если данная опция активна, <code>findface-video-worker</code> помещает лишние кадры в очередь, чтобы обработать их впоследствии. Значение по умолчанию: false.
ffmpeg_format	Передаёт формат FFmpeg (mxf, flv и т. д.), если он не может быть автоматически определен.
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.
filter_max_facesize	Указывает максимальный размер лица в пикселях. Лица большего размера не отправляются. Значение по умолчанию: 8192.
filter_min_facesize	Указывает минимальный размер лица в пикселях. Лица меньшего размера не отправляются. Значение по умолчанию: 1.
filter_min_quality	Указывает минимальное пороговое значение качества изображения лица. Значение по умолчанию: 0.45. Не меняйте значение по умолчанию без предварительной консультации с нашими техническими экспертами (support@ntechlab.com).
imotion_threshold	Минимальная интенсивность движения, которая будет регистрироваться детектором движения. Пороговое значение определяется эмпирически. Реперные точки: 0 = детектор выключен, 0.002 = значение по умолчанию, 0.05 = минимальная интенсивность слишком высока, чтобы зарегистрировать движение.
jpeg_quality	Качество сжатия исходного кадра в JPEG. Значение по умолчанию: 95 % от исходного размера.
normalized_only	Включает/отключает отправку только нормализованных лиц без исходных кадров. Значение по умолчанию: false.
overall_only	Буферный режим. Отправлять для лица один кадр наилучшего качества. Значение по умолчанию: true.
play_speed	Если меньше нуля, то скорость не ограничивается. В остальных случаях поток читается со скоростью <code>play_speed</code> . Не применимо для потоков с камер видеонаблюдения.
post_first_trackframe	Отправлять первый кадр трека. Значение по умолчанию: false.
post_last_trackframe	Отправлять последний кадр трека. Значение по умолчанию: false.
realtime_post_trackinginterval	Интервал реального времени. Включает отправку лучшего кадра в течение периода времени <code>realtime_post_interval</code> . Если false, лучший кадр ищется динамически в порядке возрастания качества. Значение по умолчанию: false.
realtime_post_trackingimmediate	Отправлять кадр сразу после его появления в поле зрения видеокамеры (режим реального времени). Значение по умолчанию: false.
realtime_post_trackinginterval	Интервал режима реального времени. Период времени в секундах, в течение которого детектор лиц выбирает лучший кадр из трека и отправляет его в компонент <code>findface-facerouter</code> . Значение по умолчанию: 1.
router_body	Массив дополнительных полей в POST-запросе с изображением лица в формате ["ключ=значение"]. По умолчанию дополнительные поля не передаются.
roi	Включает отправку на Сервер лиц, обнаруженных только внутри интересующей области $W \times H + X + Y$. По умолчанию область не задана.
rot	Включает детектирование и отслеживание лиц только внутри заданной прямоугольной области $W \times H + X + Y$. Используйте данную опцию, чтобы уменьшить нагрузку на <code>findface-video-worker</code> . По умолчанию область не задана.
router_disable_normalized	Отправлять только полные кадры без нормализованных изображений. Не включайте этот параметр, не посоветовавшись с нашей командой, так как он может повлиять на работу всей системы. Значение по умолчанию: false

1. Если вы выбрали пакет `findface-video-worker-cpu` с ускорением на CPU, используйте файл конфигурации `/etc/findface-video-worker-cpu.ini`:

```

## read streams from file, do not use VideoManager
## type:string env:CFG_INPUT longopt:--input
input =

## exit on first finished job, only when --input specified
## type:bool env:CFG_EXIT_ON_FIRST_FINISHED longopt:--exit-on-first-finished
exit_on_first_finished = false

## batch size
## type:number env:CFG_BATCH_SIZE longopt:--batch-size
batch_size = 4

## http server port for metrics, 0=do not start server
## type:number env:CFG_METRICS_PORT longopt:--metrics-port
metrics_port = 0

## resize scale, 1=do not resize
## type:double env:CFG_RESIZE_SCALE longopt:--resize-scale
resize_scale = 1.000000

## maximum number of streams
## type:number env:CFG_CAPACITY longopt:--capacity
capacity = 10

## command to obtain videomanager's grpc ip:port
## type:string env:CFG_MGR_CMD longopt:--mgr-cmd
mgr_cmd =

## videomanager grpc ip:port
## type:string env:CFG_MGR_STATIC longopt:--mgr-static
mgr_static = 127.0.0.1:18811

## ntlS server ip:port
## type:string env:CFG_NTLS_ADDR longopt:--ntls-addr
ntls_addr = 127.0.0.1:3133

## debug: save faces to dir
## type:string env:CFG_SAVE_DIR longopt:--save-dir
save_dir =

## minimum face size
## type:number env:CFG_MIN_FACE_SIZE longopt:--min-face-size
min_face_size = 60

## preinit detector for specified resolutions: "640x480;1920x1080"
## type:string env:CFG_RESOLUTIONS longopt:--resolutions
resolutions =

## use `resolutions` as only possible values, others will rescale
## type:bool env:CFG_STRICT_RESOLUTIONS longopt:--strict-resolutions

```

(continues on next page)

(продолжение с предыдущей страницы)

```
strict_resolutions = false

## worker labels: labels = k=v;group=enter
## type:string env:CFG_LABELS longopt:--labels
labels =

## use timestamps from SEI packet
## type:bool env:CFG_USE_TIME_FROM_SEI longopt:--use-time-from-sei
use_time_from_sei = false

## reader frame buffer size
## type:number env:CFG_FRAME_BUFFER_SIZE longopt:--frame-buffer-size
frame_buffer_size = 128

## skip count
## type:number env:CFG_SKIP_COUNT longopt:--skip-count
skip_count = 2

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

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

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

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

#-----
[liveness]
#-----
## path to liveness fnk
## type:string env:CFG_LIVENESS_FNK longopt:--liveness-fnk
fnk =

## path to normalization for liveness
## type:string env:CFG_LIVENESS_NORM longopt:--liveness-norm
norm =

## liveness internal algo param
## type:double env:CFG_LIVENESS_INTERVAL longopt:--liveness-interval
interval = 1.000000
```

(continues on next page)

(продолжение с предыдущей страницы)

```
## liveness internal algo param
## type:number env:CFG_LIVENESS_STDEV_CNT longopt:--liveness-stdev-cnt
stdev_cnt = 1

#-----
[imotion]
#-----
## use shared decoder for imotion (experimental)
## type:bool env:CFG_IMOTION_SHARED_DECODER longopt:--imotion-shared-decoder
shared_decoder = false

#-----
[send]
#-----
## posting faces threads
## type:number env:CFG_SEND_THREADS longopt:--send-threads
threads = 8

## posting faces maximum queue size
## type:number env:CFG_SEND_QUEUE_LIMIT longopt:--send-queue-limit
queue_limit = 256

#-----
[tracker]
#-----
## interpolate undetected bboxes in track
## type:bool env:CFG_TRACKER_INTERPOLATE_BBOXES longopt:--tracker-interpolate-bboxes
interpolate_bboxes = true

## max face miss duration, sec
## type:double env:CFG_TRACKER_MISS_INTERVAL longopt:--tracker-miss-interval
miss_interval = 1.000000

## overlap threshold
## type:double env:CFG_TRACKER_OVERLAP_THRESHOLD longopt:--tracker-overlap-threshold
overlap_threshold = 0.250000

#-----
[models]
#-----
## path to pedestrian detector fnk
## type:string env:CFG_MODELS_BODY_DETECTOR longopt:--models-body-detector
body_detector =

## path to detector fnk
## type:string env:CFG_MODELS_DETECTOR longopt:--models-detector
detector = /usr/share/findface-data/models/facedet/cheetah_fast.cpu.fnk

## path to quality fnk
## type:string env:CFG_MODELS_QUALITY longopt:--models-quality
quality = /usr/share/findface-data/models/faceattr/quality.v1.cpu.fnk
```

(continues on next page)

(продолжение с предыдущей страницы)

```
## path to norm for quality fnk
## type:string env:CFG_MODELS_NORM_QUALITY longopt:--models-norm-quality
norm_quality = /usr/share/findface-data/models/facenorm/bee_fast.cpu.fnk

## path to norm_crop2x fnk, for face send
## type:string env:CFG_MODELS_NORM_CROP2X longopt:--models-norm-crop2x
norm_crop2x = /usr/share/findface-data/models/facenorm/crop2x.v2_maxsize400.cpu.fnk

## path to cache directory
## type:string env:CFG_MODELS_CACHE_DIR longopt:--models-cache-dir
cache_dir = /var/cache/findface/models_cache
```

Если вы выбрали пакет `findface-video-worker-gpu` с ускорением на GPU, используйте файл конфигурации `/etc/findface-video-worker-gpu.ini`.

```
## cuda device number
## type:number env:CFG_DEVICE_NUMBER longopt:--device-number
device_number = 0

## read streams from file, do not use VideoManager
## type:string env:CFG_INPUT longopt:--input
input =

## exit on first finished job, only when --input specified
## type:bool env:CFG_EXIT_ON_FIRST_FINISHED longopt:--exit-on-first-finished
exit_on_first_finished = false

## batch size
## type:number env:CFG_BATCH_SIZE longopt:--batch-size
batch_size = 8

## http server port for metrics, 0=do not start server
## type:number env:CFG_METRICS_PORT longopt:--metrics-port
metrics_port = 0

## resize scale, 1=do not resize
## type:double env:CFG_RESIZE_SCALE longopt:--resize-scale
resize_scale = 1.000000

## maximum number of streams
## type:number env:CFG_CAPACITY longopt:--capacity
capacity = 30

## command to obtain videomanager's grpc ip:port
## type:string env:CFG_MGR_CMD longopt:--mgr-cmd
mgr_cmd =

## videomanager grpc ip:port
## type:string env:CFG_MGR_STATIC longopt:--mgr-static
mgr_static = 127.0.0.1:18811

## ntlс server ip:port
```

(continues on next page)

(продолжение с предыдущей страницы)

```

## type:string env:CFG_NTLS_ADDR longopt:--ntls-addr
ntls_addr = 127.0.0.1:3133

## debug: save faces to dir
## type:string env:CFG_SAVE_DIR longopt:--save-dir
save_dir =

## minimum face size
## type:number env:CFG_MIN_FACE_SIZE longopt:--min-face-size
min_face_size = 60

## preinit detector for specified resolutions: "640x480;1920x1080"
## type:string env:CFG_RESOLUTIONS longopt:--resolutions
resolutions =

## use `resolutions` as only possible values, others will rescale
## type:bool env:CFG_STRICT_RESOLUTIONS longopt:--strict-resolutions
strict_resolutions = false

## worker labels: labels = k=v;group=enter
## type:string env:CFG_LABELS longopt:--labels
labels =

## use timestamps from SEI packet
## type:bool env:CFG_USE_TIME_FROM_SEI longopt:--use-time-from-sei
use_time_from_sei = false

## reader frame buffer size
## type:number env:CFG_FRAME_BUFFER_SIZE longopt:--frame-buffer-size
frame_buffer_size = 128

## skip count
## type:number env:CFG_SKIP_COUNT longopt:--skip-count
skip_count = 0

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

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

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

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

```

(continues on next page)

(продолжение с предыдущей страницы)

```
tracks_last = false

#-----
[liveness]
#-----
## path to liveness fnk
## type:string env:CFG_LIVENESS_FNK longopt:--liveness-fnk
fnk =

## path to normalization for liveness
## type:string env:CFG_LIVENESS_NORM longopt:--liveness-norm
norm =

## liveness internal algo param
## type:double env:CFG_LIVENESS_INTERVAL longopt:--liveness-interval
interval = 1.000000

## liveness internal algo param
## type:number env:CFG_LIVENESS_STDEV_CNT longopt:--liveness-stdev-cnt
stdev_cnt = 1

#-----
[imotion]
#-----
## use shared decoder for imotion (experimental)
## type:bool env:CFG_IMOTION_SHARED_DECODER longopt:--imotion-shared-decoder
shared_decoder = false

#-----
[send]
#-----
## posting faces threads
## type:number env:CFG_SEND_THREADS longopt:--send-threads
threads = 8

## posting faces maximum queue size
## type:number env:CFG_SEND_QUEUE_LIMIT longopt:--send-queue-limit
queue_limit = 256

#-----
[tracker]
#-----
## interpolate undetected bboxes in track
## type:bool env:CFG_TRACKER_INTERPOLATE_BBOXES longopt:--tracker-interpolate-bboxes
interpolate_bboxes = true

## max face miss duration, sec
## type:double env:CFG_TRACKER_MISS_INTERVAL longopt:--tracker-miss-interval
miss_interval = 1.000000

## overlap threshold
## type:double env:CFG_TRACKER_OVERLAP_THRESHOLD longopt:--tracker-overlap-threshold
```

(continues on next page)

(продолжение с предыдущей страницы)

```
overlap_threshold = 0.250000

#-----
[models]
#-----
## path to pedestrian detector fnk
## type:string env:CFG_MODELS_BODY_DETECTOR longopt:--models-body-detector
body_detector =

## path to detector fnk
## type:string env:CFG_MODELS_DETECTOR longopt:--models-detector
detector = /usr/share/findface-data/models/facedet/cheetah_fast.gpu.fnk

## path to quality fnk
## type:string env:CFG_MODELS_QUALITY longopt:--models-quality
quality = /usr/share/findface-data/models/faceattr/quality.v1.gpu.fnk

## path to norm for quality fnk
## type:string env:CFG_MODELS_NORM_QUALITY longopt:--models-norm-quality
norm_quality = /usr/share/findface-data/models/facenorm/bee_fast.gpu.fnk

## path to norm_crop2x fnk, for face send
## type:string env:CFG_MODELS_NORM_CROP2X longopt:--models-norm-crop2x
norm_crop2x = /usr/share/findface-data/models/facenorm/crop2x.v2_maxsize400.gpu.fnk

## path to cache directory
## type:string env:CFG_MODELS_CACHE_DIR longopt:--models-cache-dir
cache_dir = /var/cache/findface/models_cache

#-----
[video_decoder]
#-----
## decode video on cpu
## type:bool env:CFG_VIDEO_DECODER_CPU longopt:--video-decoder-cpu
cpu = false
```

Пользовательская настройка findface-video-worker на CPU/GPU выполняется с использованием следующих параметров:

CPU	GPU	Описание
batch_size		Отправлять лица в пакетах указанного размера.
capacity		Максимальное количество видеопотоков, обрабатываемых findface-video-worker.
Н/п	cpu	При необходимости декодировать видео на CPU.
Н/п	device_number	Номер используемого GPU-устройства.
exit_on_first_finished		Выход (если указан input) Выйти после завершения первого job-задания.
input		Обрабатывать видеопотоки из файла, игнорируя данные потоков, поступающие от findface-video-manager.
labels		Метки, используемые для привязки экземпляра findface-video-worker к определенной группе камер. См. <i>Привязка группы камер к экземпляру findface-video-worker</i> .
liveness -> fnk		Путь к детектору <i>живых</i> лиц (Liveness).
liveness -> norm		Путь к нормализатору, используемому в детекторе <i>живых</i> лиц (Liveness).
mgr-cmd		Команда для получения IP-адреса компонента findface-video-manager (опционально вместо mgr-static).
mgr-static		IP-адрес сервера findface-video-manager, который обеспечивает findface-video-worker настройками и списком видеопотоков для обработки.
metrics_port		Порт HTTP-сервера для отправки метрик. Если 0, метрики не отправляются.
min_face_size		Минимальный обнаруживаемый размер лица.
ntls-addr		IP-адрес и порт сервера findface-ntls.
resize_scale		Масштабировать видеокадры с заданным коэффициентом.
resolutions		Предварительно инициализируйте findface-video-worker для конкретных разрешений. Например, «640x480;1920x1080».
save_dir		(Для отладки) Сохранять обнаруженные лица в заданный каталог.
streamer -> port, url		IP-адрес и порт для доступа к <i>видеостене</i> .
use_time_from_sei		(Для MPEG-2) Используйте временные метки SEI (supplemental enhancement information).

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.

Существует 2 типа job-заданий:

- камера:

```
curl http://127.0.0.1:18810/job/ffsec-camera:22 | jq
{
  "id": "ffsec-camera:22",
  "enabled": true,
  "stream_url": "rtsp://ntech:Ntech11@172.20.77.33:654/
↪000000010000babe0000accc8e9e3a58/live",
  "labels": {},
  "router_url": "http://127.0.0.1/video-detector/frame",
  "single_pass": false,
  "stream_settings": {
```

(continues on next page)

(продолжение с предыдущей страницы)

```

"ffmpeg_params": [],
"md_threshold": 0.002,
"md_scale": 0.3,
"fd_frame_height": -1,
"uc_max_time_diff": 30,
"uc_max_dup": 3,
"uc_max_avg_shift": 10,
"det_period": 8,
"realtime": false,
"npersons": 4,
"disable_drops": false,
"tracker_threads": 4,
"parse_sei": false,
"image_arg": "photo",
"additional_headers": [],
"additional_body": [],
"api_timeout": 15000,
"api_ssl_verify": true,
"post_uniq": true,
"min_score": -2,
"min_d_score": -1000,
"realtime_dly": 500,
"realtime_post_perm": false,
"rot": "",
"roi": "",
"draw_track": false,
"send_track": 0,
"min_face_size": 0,
"max_face_size": 0,
"overall": true,
"only_norm": false,
"max_candidates": 0,
"jpeg_quality": 95,
"ffmpeg_format": ""
},
"stream_settings_gpu": {
  "play_speed": -1,
  "filter_min_quality": 0.45,
  "filter_min_face_size": 1,
  "filter_max_face_size": 8192,
  "normalized_only": false,
  "jpeg_quality": 95,
  "overall_only": false,
  "use_stream_timestamp": false,
  "ffmpeg_params": [],
  "router_timeout_ms": 15000,
  "router_verify_ssl": true,
  "router_headers": [
    "Authorization=Token 7db297c4107518b52b4e2195b72c5947"
  ],
  "router_body": [],
  "start_stream_timestamp": 0,

```

(continues on next page)

(продолжение с предыдущей страницы)

```

"imotion_threshold": 0,
"rot": "",
"roi": "",
"realtime_post_interval": 1,
"realtime_post_every_interval": false,
"ffmpeg_format": "",
"disable_drops": false,
"router_full_frame_png": false,
"router_disable_normalized": false,
"crop_fullframe_rot": false,
"realtime_post_first_immediately": false
},
"status": "INPROGRESS",
"status_msg": "",
"statistic": {
  "processed_duration": 5729.5728,
  "faces_posted": 43,
  "faces_failed": 0,
  "faces_not_posted": 0,
  "processing_fps": 15.589469,
  "frames_dropped": 0,
  "frames_processed": 87121,
  "frames_imotion_skipped": 0,
  "decoding_soft_errors": 0,
  "frame_width": 1920,
  "frame_height": 1200,
  "job_starts": 1
},
"restream_url": "ws://127.0.0.1:18999/stream/ffsec-camera:22",
"shots_url": "http://127.0.0.1:18999/shot/ffsec-camera:22",
"worker_id": "tevmenova-ntechlab_gpu_2a1fd5290195670b689d8e01e93b673c",
"version": "bt1s8kjm51nv2hqc4qg0"

```

- видеоархив:

```

curl http://127.0.0.1:18810/job/ffsec-video-archive:1 | jq
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 2094    0 2094    0    0   2044k    0  --:--:--  --:--:--  --:--:-- 2044k
{
  "id": "ffsec-video-archive:1",
  "enabled": true,
  "stream_url": "file:///var/lib/findface-security/uploads/videos/1.bin",
  "labels": {
    "camera_group_id": "1",
    "camera_id": ""
  },
  "router_url": "http://127.0.0.1/video-detector/frame",
  "single_pass": true,
  "stream_settings": {
    "ffmpeg_params": [],
    "md_threshold": 0.002,

```

(continues on next page)

(продолжение с предыдущей страницы)

```

"md_scale": 0.3,
"fd_frame_height": -1,
"uc_max_time_diff": 30,
"uc_max_dup": 3,
"uc_max_avg_shift": 10,
"det_period": 8,
"realtime": false,
"npersons": 4,
"disable_drops": false,
"tracker_threads": 4,
"parse_sei": false,
"image_arg": "photo",
"additional_headers": [],
"additional_body": [],
"api_timeout": 15000,
"api_ssl_verify": true,
"post_uniq": true,
"min_score": -2,
"min_d_score": -1000,
"realtime_dly": 500,
"realtime_post_perm": false,
"rot": "",
"roi": "",
"draw_track": false,
"send_track": 0,
"min_face_size": 0,
"max_face_size": 0,
"overall": true,
"only_norm": false,
"max_candidates": 0,
"jpeg_quality": 95,
"ffmpeg_format": ""
},
"stream_settings_gpu": {
  "play_speed": -1,
  "filter_min_quality": 0.45,
  "filter_min_face_size": 1,
  "filter_max_face_size": 8192,
  "normalized_only": false,
  "jpeg_quality": 95,
  "overall_only": false,
  "use_stream_timestamp": false,
  "ffmpeg_params": [],
  "router_timeout_ms": 15000,
  "router_verify_ssl": true,
  "router_headers": [
    "Authorization=Token 7db297c4107518b52b4e2195b72c5947"
  ],
  "router_body": [],
  "start_stream_timestamp": 0,
  "imotion_threshold": 0,
  "rot": ""
}

```

(continues on next page)

(продолжение с предыдущей страницы)

```

"roi": "",
"realtime_post_interval": 1,
"realtime_post_every_interval": false,
"ffmpeg_format": "",
"disable_drops": true,
"router_full_frame_png": false,
"router_disable_normalized": false,
"crop_fullframe_rot": false,
"realtime_post_first_immediately": false
},
"status": "INPROGRESS",
"status_msg": "",
"statistic": {
  "processed_duration": 291,
  "faces_posted": 335,
  "faces_failed": 0,
  "faces_not_posted": 0,
  "processing_fps": 359.69928,
  "frames_dropped": 0,
  "frames_processed": 8731,
  "frames_imotion_skipped": 0,
  "decoding_soft_errors": 0,
  "frame_width": 1280,
  "frame_height": 720,
  "job_starts": 1
},
"restream_url": "ws://127.0.0.1:18999/stream/ffsec-video-archive:1",
"shots_url": "http://127.0.0.1:18999/shot/ffsec-video-archive:1",
"worker_id": "tevmenova-ntechlab_gpu_2a1fd5290195670b689d8e01e93b673c",
"version": "bt1tm3bm51nv2hqc4qh0"

```

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 ПК FindFace).
- **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 ПК FindFace), в который компонент `findface-video-worker` отправляет обнаруженные лица для последующей обработки.
- **stream_settings**: используется только для обратной совместимости.
- **stream_settings_gpu**: 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 лиц, 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.

Для управления `findface-ntls` используйте основной веб-интерфейс ПК FindFace. Доступны следующие возможности:

- 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 работы компонента составляет 6 часов.

Настройка компонента `findface-ntls` выполняется через файл конфигурации `/etc/findface-ntls.cfg`.

```
## Address to accept incoming client connections (IP:PORT)
## type:string env:CFG_LISTEN longopt:--listen
listen = 127.0.0.1:3133
## Directory where license files are stored
## type:string env:CFG_LICENSE_DIR longopt:--license-dir
license_dir = /opt/ntech/license
## Use specified proxy (MUST support HTTP CONNECT method) to access global license_
↪server (IP:PORT)
## type:string env:CFG_PROXY longopt:--proxy
proxy =
## Bind address for embedded UI (IP:PORT)
## type:string env:CFG_UI longopt:--ui
ui = 127.0.0.1:3185
```

Пользовательская настройка `findface-ntls` выполняется с использованием следующих параметров:

Параметр	Описание
<code>license_dir</code>	Каталог 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 лиц. Он получает рамку с лицом и нормализованное изображение лица 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`.

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

#
↳ =====
# FindFace Security configuration file
#
↳ =====
#
# This config file is written in Python's syntax and interpreted at FindFace
↳ Security
# service startup. You have to restart the service in order to apply changes.
#
# If you have any questions or suggestions, please contact us at
↳ support@ntechlab.com

#
↳ =====
# GENERAL SETTINGS
#
↳ =====

# enables additional logs
DEBUG = False
```

(continues on next page)

(продолжение с предыдущей страницы)

```
# media files directory
MEDIA_ROOT = "/var/lib/findface-security/uploads"

# static files directory
STATIC_ROOT = "/var/lib/findface-security/static"

# language code
LANGUAGE_CODE = 'en-us'

# time zone
TIME_ZONE = 'UTC'

# 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':
        ↪ 'ZnUqeWKMzT6T2Bj2G4gbFn7cxRSMcxS7'
    }
}

# Signature key for session encryption
# Use pugen -sncy 50 1/tr "" "." to generate your own unique key
SECRET_KEY = '8b26839acde2633362bdb176e741a650'

#_
↪ =====
# FINDFACE SECURITY SETTINGS
#_
↪ =====

# 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://172.20.77.120'

# EXTERNAL_ADDRESS is used to access objects created inside FFSecurity via_
↪ external links.
EXTERNAL_ADDRESS = ''

# - Base FFSecurity settings -

# enable permissions system
ENABLE_ACL = True

FFSECURITY = {
    # findface-video-worker authorization token
```

(continues on next page)

(продолжение с предыдущей страницы)

```
'VIDEO_DETECTOR_TOKEN': '3243a92b03c3411d4faa3cdd72f967b6',

# base face matching confidence threshold
'CONFIDENCE_THRESHOLD': 0.745,

# episodes specific matching threshold that is used to join faces in an
→episode
'EPISODES_THRESHOLD': 0.689,

# minimum face quality sufficient to add it to a dossier
'MINIMUM_DOSSIER_QUALITY': 0.45,

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

# 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
'BLURRED_FULLFRAME_JPEG_QUALITY': 85,

# matched events older than EVENTS_MAX_MATCHED_AGE will be automatically
# deleted (every night at 1:17 am by default)
'EVENTS_MAX_MATCHED_AGE': 30,

# same as above but for unmatched events
'EVENTS_MAX_UNMATCHED_AGE': 30,

# same as EVENTS_MAX_MATCHED_AGE but for matched full frame images only
→(thumbnails won't be deleted)
'EVENTS_MAX_FULLFRAME_UNMATCHED_AGE': 30,

# same as above but for unmatched full frame images only (thumbnails won't
→be deleted)
'EVENTS_MAX_FULLFRAME_MATCHED_AGE': 30,

# same as above but for counter records
'COUNTER_RECORDS_MAX_AGE': 30,

# same as above but for person events (if no person events left in person,
→it is deleted too)
'PERSON_EVENTS_MAX_AGE': 90,

# when closing episode, delete all events except the best episode event
'EPISODE_KEEP_ONLY_BEST_EVENT': False,

# NTLs licence server url
'NTLS_HTTP_URL': 'http://127.0.0.1:3185',

# findface-video-worker face posting address,
# it must be set to either FFSecurity EXTERNAL_ADDRESS (by default)
```

(continues on next page)

(продолжение с предыдущей страницы)

```
# or findface-facerouter url (in some specific cases)
'ROUTER_URL': 'http://127.0.0.1',

# send serialized dossiers, dossier-lists, camera and camera groups in
↳webhooks
'VERBOSE_WEBHOOKS': False,

# jpeg quality used when saving thumbnails
'THUMBNAIL_JPEG_QUALITY': 75,

# FFServer services urls
'VIDEO_MANAGER_ADDRESS': 'http://127.0.0.1:18810',
'SF_API_ADDRESS': 'http://127.0.0.1:18411',
'FFCOUNTER_ADDRESS': 'http://127.0.0.1:18300',
'LIVENESS_API_ADDRESS': 'http://127.0.0.1:18301',

# upload video archives to this path, it differs from media root and
# you have to change alias in corresponding nginx location also (/videos/)
'VIDEO_ARCHIVE_UPLOAD_PATH': '/var/lib/findface-security/uploads/videos/',

# additional events features.
# make sure that corresponding extractors
# are licensed and enabled at findface-extraction-api config file.
# available features are: gender, age, emotions, beard, glasses, medmask.
'EVENTS_FEATURES': [],

# feature specific confidence thresholds
'LIVENESS_THRESHOLD': 0.85,
'EMOTIONS_THRESHOLD': 0.25,
'BEARD_THRESHOLD': 0.7,

# counters full frame saving options:
# `always` - save always
# `detect` - save only if faces or silhouettes have been detected
# `never` - never save full frames
'COUNTERS_SAVE_FULLFRAME': 'always',
'COUNTERS_FULLFRAME_JPEG_QUALITY': 75,
'COUNTERS_THUMBNAIL_JPEG_QUALITY': 75,

# max camera frames_dropped percent
'MAX_CAMERA_DROPPED_FRAMES': {'yellow': 0.1, 'red': 0.3},
# max camera faces_failed percent
'MAX_CAMERA_FAILED_FACES': {'yellow': 0.1, 'red': 0.3},

# -- 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
```

(continues on next page)

(продолжение с предыдущей страницы)

```
'PERSONS_CONFIDENCE_THRESHOLD': 0.745,

# 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,

# maximum concurrent video manager jobs for video archives processing
'MAX_VIDEO_ARCHIVE_JOBS': 3,

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

# -- Startup tests --

# required services availability test
'SERVICES_AVAILABILITY_TEST': True,

# enable saving audit logs to PostgreSQL
'ENABLE_AUDIT_LOGS': True,

# -- FFSecurity Onvif --

# auth credentials for ffsecurity_onvif
# ONVIF_CREDENTIALS = [
#     {
#         "hostnames": ["192.168.1.64",
↪ "2a00:1370:8117:ab87:a614:37ff:fe49:2683"],
#         "login": "admin",
#         "password": "admin123"
#     }
# ],
'ONVIF_CREDENTIALS': {},
# list of all hostnames that will be ignored during Onvif service discovery
# ONVIF_IGNORE_LIST = ["192.168.1.217"],
'ONVIF_IGNORE_LIST': [],

# -- Optional parameters --

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

# Edit CUSTOM_FIELDS->dossier_face section to customize dossier face
↪ content.
# Below is an example with every field type possible.

# 'CUSTOM_FIELDS': {
#     'dossier_meta': {
```

(continues on next page)

(продолжение с предыдущей страницы)

```

#       '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',
#               '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'
#           }
#       ]
#   },
#   'dossier_face': {
#       '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",

```

(continues on next page)

(продолжение с предыдущей страницы)

```

#           "type": "bool",
#           "default": True
#       },
#   ]
# }
# },

# maximum event age in seconds than could be added to an episode.
# 'EPISODE_SEARCH_INTERVAL': 60,
# If none of these events matched, new episode is created.

# maximum episode duration (episode is closed after)
# 'EPISODE_MAX_DURATION': 300,

# if no new event added to an episode during this timeout, episode will be
→closed.
# 'EPISODE_EVENT_TIMEOUT': 30,

# maximum created thumbnail width
# 'THUMBNAIL_MAX_WIDTH': 320,

# url of the backend which is used for social network search.
# contact support for additional information.
# 'SOCIAL_BACKEND': None,

# additional social backend headers.
# 'SOCIAL_HEADERS': {},

# unacknowledged events notification interval
# 'UNACKNOWLEDGED_NOTIFY_INTERVAL': 1,

# set to True to run all media requests (photos, attachments) through the
# django application for acl checks.
# enabling this setting slightly increases security but
# has severe negative effects on performance.
# you will also have to mark /uploads/ location as 'internal' in nginx
→config
#
# 'OVERPROTECT_MEDIA': False,
}

# - FindFace Security authorization configuration dictionary -

FFSECURITY_AUTH_CONFIG = {
# available options: face, password, face_and_password, face_or_password
'AUTH_TYPE': 'face_or_password',
'FACE_AUTH_CONFIDENCE': 0.745,
# 180 days by default
'MAXIMUM_SESSION_LENGTH': 15552000,
# session renew works only with face or face_or_password authorization type
'NEED_SESSION_RENEW': False,
'RENEW_SESSION_INTERVAL': 0,

```

(continues on next page)

(продолжение с предыдущей страницы)

```

'MAXIMUM_RENEW_ATTEMPTS': 2,
}

# - FindFace Security user interface configuration dictionary -

FFSECURITY_UI_CONFIG = {
    "event": {
        "features": {
            "f_gender_class": ["male", "female"],
            "age": {
                "f_age_gte": "",
                "f_age_lte": ""
            },
            "f_emotions_class": ["angry", "disgust", "fear", "happy", "sad",
↪ "surprise", "neutral"],
            "f_glasses_class": ["none", "eye", "sun"],
            "f_beard_class": ["none", "beard"],
            "f_liveness_class": ["real", "fake"],
            "f_medmask_class": ["none", "incorrect", "correct"],
        }
    },
},

# Adjustable confidence threshold presets for face matching.
# Please consult with our support before changing.
"confidence_display": [
    {"confidence": 0.00, "color": "#000000", "label": {"ru": "Очень низкий
↪", "en": "Very Low"}},
    {"confidence": 0.65, "color": "#FF0300", "label": {"ru": "Низкий", "en
↪": "Low"}},
    {"confidence": 0.70, "color": "#FFB700", "label": {"ru": "Пониженный",
↪ "en": "Slightly Low"}},
    {"confidence": 0.72, "color": "#B8FA00", "label": {"ru": "Нормальный",
↪ "en": "Moderate"}},
    {"confidence": 0.75, "color": "#7EFF2B", "label": {"ru": "Повышенный",
↪ "en": "Slightly High"}},
    {"confidence": 0.80, "color": "#4DFF60", "label": {"ru": "Высокий", "en
↪": "High"}},
    {"confidence": 0.85, "color": "#1DFF96", "label": {"ru": "Очень высокий
↪", "en": "Very High"}},
]
}

# -- ASGI-server configuration --
# consult support before changing these settings.

# per worker thread pool size.
ASGI_THREADS = 32

UVICORN_SETTINGS = {
    # worker processes count, 'auto' sets it to logical cpu count
    'workers': 'auto',
    'host': 'localhost',
}

```

(continues on next page)

(продолжение с предыдущей страницы)

```

'port': 8002,
# websocket worker processes count,
# 'auto' sets it to logical cpu count, but not more than 8.
'ws-workers': 'auto',
'ws-host': 'localhost',
'ws-port': 8003,
}

# 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,
    }
}

# -- Other settings --

# The number of threads in the night clusterization.
# Not recommended values are greater than the number of cores in the processor.
# Consult with support before changing this value.
NUMPY_OMP_NUM_THREADS = 'auto'

#_
↪=====
# FINDFACE SECURITY PLUGINS
#_
↪=====
# Uncomment lines below to enable plugins. Please consult documentation for
# a plugin specific settings.

# ===== Axxon =====
# INSTALLED_APPS.append('ffsecurity_axxon')

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

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

```

(continues on next page)

(продолжение с предыдущей страницы)

```
# ===== Genetec =====
# INSTALLED_APPS.append('ffsecurity_genetec')

# ===== Sova =====
# INSTALLED_APPS.append('ffsecurity_sova')

# ===== Sigur =====
# keep in mind, that SIGUR plugin also uses CUSTOM_FIELDS and THUMBNAI_MAX_
↳WIDTH settings
# INSTALLED_APPS.append('ffsecurity_sigur')
# SIGUR = {
#     'LOGIN': 'admin',
#     'PASSWORD': 'admin',
#     'MF_SELECTOR': 'biggest', # what to do with several faces in sigur_
↳person photo; allowed ['biggest', 'reject']
#     'ONLY_RT_EVENTS': True, # only events with bs_type == realtime,
#     'EVENT_DELAY': 0.004 # minimum time between two events of same person in_
↳seconds. If interval between two events with same person is less, than this_
↳value, second event will be dropped
# }

# ===== CryptoPRO authentication =====
# INSTALLED_APPS.append('ffsecurity_cproauth')
# REST_FRAMEWORK['DEFAULT_AUTHENTICATION_CLASSES'] = [
#     'ffsecurity.auth.TokenAuthentication',
#     'ffsecurity_cproauth.auth.CryptoProOrTokenAuthentication'
# ]

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

# token must be identical on master and slave
# use pwgen -s 64 1
# SYNC_TOKEN = 'change_me'
# rrule that defines sync schedule
# SYNC_SCHEDULE = 'RRULE:FREQ=DAILY;WKST=MO;BYHOUR=4;BYMINUTE=0'
# if True synchronization will occur on FindFace Security startup and restart
# SYNC_AT_STARTUP = False
# if True synchronization will occur immediately after creating_
↳synchronization for dossier list
# SYNC_AT_CREATION = False

# ===== Puppeteer =====
# INSTALLED_APPS.append('ffsecurity_puppeteer')
```

(continues on next page)

(продолжение с предыдущей страницы)

```
# PUPPETEER_CONFIG = {
#   'UNSAVED_RESULTS_DELETION_TIMEOUT': 3600,           # maximum lifetime of
#   ↳search results not saved involuntarily
#   'REMOTE_MONITORING_SYNC_INTERVAL': 600,           # monitoring data
#   ↳synchronization interval, seconds
#   'ENABLE_DAILY_SEARCH': True,                       # daily search
#   ↳activation (default False)
#   'DAILY_SEARCH_PUSH_HOUR': 2,                       # daily search dossiers
#   ↳synchronization hour
#   'DAILY_SEARCH_PULL_HOUR': 6,                       # hour in which results
#   ↳of daily search will be obtained
#   'puppets': [
#     {
#       'id': 'first_puppet',                           # puppet ID
#       'url': 'http://1.1.1.1:8010/',                 # puppet URL
#       'token': 'first_puppet_token',                 # use pwgen -s 64 1
#       ↳(should match the token in puppet)
#       'facen_model': 'jackfruit_480'                 # face model in puppet
#     },
#     {
#       'id': 'second_puppet',
#       'url': 'http://1.1.1.1:8010/',
#       'token': 'second_puppet_token',
#
#       # if remote installation has a different face model than the
#       ↳one used in FFSecurity -
#       # you need to specify its name and ExtractionAPI URL where the
#       ↳corresponding face model is specified
#       'facen_model': 'grapefruit_480',
#       'extractor': 'http://127.0.0.1:18667',
#     },
#   ]
# }
#
# ===== Vns =====
# A plugin for using FindFace Security as a puppeteer server
# INSTALLED_APPS.append('ffsecurity_vns')
#
# VNS_CONFIG = {
#   'USERS': {
#     'user1': 'token1',
#     'user2': 'token2'
#   },
#   'MONITORING_THRESHOLD': 0.75,
#   'DAILY': {
#     'ENABLED': False,
#     'THRESHOLD': 0.75,
#     'START_TIME': "00:00:00"
#   }
# }
```

Пользовательская настройка findface-security выполняется с использованием следующих параметров:

Параметр	Описание
BEARD_THRESHOLD	Присутствие на лице бороды оценивается с определенной достоверностью.
CONFIDENCE_THRESHOLD	Пороговая степень схожести для верификации лиц в событиях.
COUNTERS_FULLFRAME_JPEG_QUALITY	Качество JPEG полных кадров в счетчиках.
COUNTER_RECORDS_MAX_AGE	Возраст записей счетчика, при достижении которого происходит их автоматическое удаление.
COUNTERS_SAVE_FULLFRAME	Параметры сохранения полных кадров при работе счетчиков: <code>always</code> (всегда), <code>never</code> (никогда), <code>if-needed</code> (если нужно).
COUNTERS_THUMBNAIL_JPEG_QUALITY	Качество JPEG миниатюр в счетчиках.
CUSTOM_FIELDS	Раскомментируйте и измените раздел, чтобы настроить содержание досье.
DATABASES (раздел)	Настройки базы данных. Заполните следующим образом: 'PORT': 5439, 'HOST': localhost, 'DBNAME': findface, 'USER': findface, 'PASSWORD': findface.
EMOTIONS_THRESHOLD	Присутствие эмоций оценивается с определенной достоверностью. В зависимости от настроек может использоваться для фильтрации событий.
EPISODE_EVENT_TIMEOUT	Максимальное время в секундах с момента добавления последнего события в эпизод.
EPISODE_KEEP_ONLY_BEST_EVENT	При закрытии эпизода удалять в нем все события за исключением лучшего.
EPISODE_MAX_DURATION	Максимальная продолжительность эпизода в секундах. По истечении этого времени эпизод будет закрыт.
EPISODE_SEARCH_INTERVAL	Период времени, предшествующий событию, в течение которого система ищет события.
EPISODES_THRESHOLD	Пороговая степень схожести для верификации лиц в эпизодах.
EVENTS_FEATURES	Перечислите здесь модели для распознавания атрибутов лица, которые вы хотите использовать.
EVENTS_MAX_FULLFRAME_MATCHED_AGE	То же, что <code>EVENTS_MAX_MATCHED_AGE</code> , только для полных кадров.
EVENTS_MAX_FULLFRAME_UNMATCHED_AGE	То же, что <code>EVENTS_MAX_UNMATCHED_AGE</code> , только для полных кадров.
EVENTS_MAX_MATCHED_AGE	Возраст события с совпадением, при достижении которого должно происходить событие.
EVENTS_MAX_UNMATCHED_AGE	Возраст события без совпадения, при достижении которого должно происходить событие.
EXTERNAL_ADDRESS	Внешний IP-адрес или URL, который будет использоваться для доступа к веб-интерфейсу.
EXTRACTION_API	IP-адрес сервера <code>findface-extraction-api</code> .
IGNORE_UNMATCHED	Отключает запись события в базу данных, если обнаруженное лицо отсутствует в базе данных.
LIVENESS_THRESHOLD	Детектор живых лиц оценивает живость лица с определенной достоверностью.
MAX_CAMERA_DROPPED_FRAMES	Цветовое представление статусов камеры (желтый и красный) на основании количества пропущенных кадров.
MAX_CAMERA_FAILED_FACES	Цветовое представление статусов камеры (желтый и красный) на основании количества неудачных попыток распознавания.
MAX_VIDEO_ARCHIVE_JOBS	Максимальное количество одновременных заданий на обработку видеоархива.
MINIMUM_DOSSIER_QUALITY	Минимальное качество лица на фотографии в досье. Если качество лица ниже, то досье не создается.
NTLS_HTTP_URL	IP-адрес сервера <code>findface-ntls</code> .
PERSONS_CLUSTERIZATION_SCHEDULE	Рекуррентное правило (RRULE), задающее расписание плановой кластеризации персон.
PERSONS_CONFIDENCE_THRESHOLD	Пороговая степень схожести для принятия решения о принадлежности лица к персоне.
PERSON_EVENT_MIN_EPISODE_EVENTS	Минимальное количество событий в эпизодах, участвующих в кластеризации персон.
PERSON_EVENT_MIN_QUALITY	Минимальное качество лиц, используемых в кластеризации персон. Не менее 0.5.
PERSON_EVENTS_MAX_AGE	Возраст событий распознавания персон, при достижении которого происходит удаление событий.
ROUTER_URL	IP-адрес сервера <code>findface-security</code> , который будет получать обнаруженные лица.
SERVICE_EXTERNAL_ADDRESS	(Опционально) IP-адрес, приоритетно используемый в вебхуках и интеграциях.
SF_API_ADDRESS	IP-адрес сервера <code>findface-sf-api</code> .
THUMBNAIL_JPEG_QUALITY	Качество JPEG миниатюр.
THUMBNAIL_MAX_WIDTH	Максимальная ширина миниатюры.
VERBOSE_WEBHOOKS	Отправлять в <i>вебхуках</i> сериализованные досье, списки наблюдения, камер и т.д.
VIDEO_DETECTOR_TOKEN	Придумайте токен и укажите его в данном параметре, чтобы авторизоваться на сервере.
VIDEO_MANAGER_ADDRESS	IP-адрес сервера <code>findface-video-manager</code> .

findface-facerouter

Важно: Компонент `findface-facerouter` не входит в состав стандартной конфигурации ПК FindFace. При необходимости используйте его для интеграции. См. *Пользовательские плагины*.

Компонент `findface-facerouter` представляет собой сервис, через который задаются правила обработки обнаруженных на видео лиц. Правила задаются в виде пользовательских плагинов.

Компонент `findface-facerouter` принимает нормализованные изображения лиц вместе с исходным кадром и другой информацией (например, датой и временем детекции) от компонента `findface-video-worker`. В общем случае позволяет обрабатывать лица произвольным способом, в том числе отправлять их напрямую в партнерское приложение. В базовой реализации перенаправляет их в компонент `findface-sf-api` для дальнейшей обработки в соответствии с заданными правилами.

Полный список функций:

- задание правил обработки обнаруженных лиц на видео,
- перенаправление обнаруженных лиц 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`.

```
# main.py options:
# debug                               = False
## debug - debug mode
# detector                             = ''
## detector - Detector to use if client fails to provide normalized face
## (nnd). Use "nnd" if you need to detect faces in such requests. Empty value
## rejects requests without normalized.
# host                                 = ''
## host - host to listen
# port                                 = 18820
## port - port to listen
# prometheus_timing_buckets           = None
## prometheus_timing_buckets - prometheus histogram buckets (python list of
## numbers, e.g. [1,2,3])
# sfapi_url                            = 'http://localhost:18411'
## sfapi_url - SF-API URL
# version                              = False
## version - print version
# plugin_dir.py options:
# plugin_dir                          = ''
## plugin_dir - Plugin directory for plugin_source='dir'
# abstract_define.py options:
# plugin_source                       = 'dir'
## plugin_source - Plugin source (dir)
# log.py options:
# log_file_max_size                   = 100000000
## log_file_max_size - max size of log files before rollover
# log_file_num_backups                = 10
## log_file_num_backups - number of log files to keep
# log_file_prefix                     = None
```

(continues on next page)

(продолжение с предыдущей страницы)

```
## log_file_prefix - Path prefix for log files. Note that if you are running
## multiple tornado processes, log_file_prefix must be different for each of
## them (e.g. include the port number)
# log_rotate_interval          = 1
## log_rotate_interval - The interval value of timed rotating
# log_rotate_mode              = 'size'
## log_rotate_mode - The mode of rotating files(time or size)
# log_rotate_when              = 'midnight'
## log_rotate_when - specify the type of TimedRotatingFileHandler interval other
## options:('S', 'M', 'H', 'D', 'W0'-'W6')
# log_to_stderr                = None
## log_to_stderr - Send log output to stderr (colorized if possible). By default
## use stderr if --log_file_prefix is not set and no other logging is
## configured.
# logging                      = 'info'
## logging - Set the Python log level. If 'none', tornado won't touch the
## logging configuration.
```

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 лиц.

1.7.2 Файл с параметрами установки

При установке ПК FindFace из инсталлятора параметры установки автоматически сохраняются в файл `/tmp/<findface-installer-*.json`. Вы можете отредактировать данный файл и использовать его при установке ПК FindFace 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 ПК FindFace на сервере с другим IP-адресом.

Пример файла с параметрами установки:

```
{
  "product": "security",
  "python3-pil.variant": "avx2",
  "ignore_nolicense": true,
  "type": "stand-alone",
  "ext_ip.bind": "0.0.0.0",
  "ext_ip.advertised": "172.20.77.19",
  "inter_ip.advertised": "127.0.0.1",
```

(continues on next page)

(продолжение с предыдущей страницы)

```
"inter_ip.bind": "127.0.0.1",
"components": [
  "findface-data",
  "memcached",
  "etcd",
  "redis",
  "postgres",
  "pgbouncer",
  "jq",
  "python3-pil",
  "findface-ntls",
  "findface-extraction-api",
  "findface-sf-api",
  "findface-counter",
  "findface-liveness-api",
  "findface-upload",
  "findface-video-manager",
  "findface-video-worker",
  "findface-security",
  "findface-tarantool-server"
],
"tnt_instances": 4,
"findface-video-worker.variant": "gpu",
"findface-extraction-api.variant": "gpu",
"findface-data.models": [
  "./findface-data-age.v1-cpu_3.0.0_all.deb",
  "./findface-data-age.v1-gpu_3.0.0_all.deb",
  "./findface-data-beard.v0-cpu_3.0.0_all.deb",
  "./findface-data-beard.v0-gpu_3.0.0_all.deb",
  "./findface-data-detector-cheetah-cpu_3.0.0_all.deb",
  "./findface-data-detector-cheetah-fast-cpu_3.0.0_all.deb",
  "./findface-data-detector-cheetah-fast-gpu_3.0.0_all.deb",
  "./findface-data-detector-cheetah-gpu_3.0.0_all.deb",
  "./findface-data-detector-mtcnn-cpu_3.0.0_all.deb",
  "./findface-data-emotions.v1-cpu_3.0.0_all.deb",
  "./findface-data-emotions.v1-gpu_3.0.0_all.deb",
  "./findface-data-gender.v2-cpu_3.0.0_all.deb",
  "./findface-data-gender.v2-gpu_3.0.0_all.deb",
  "./findface-data-glasses3.v0-cpu_3.0.0_all.deb",
  "./findface-data-glasses3.v0-gpu_3.0.0_all.deb",
  "./findface-data-ifruit-320-cpu_3.0.0_all.deb",
  "./findface-data-ifruit-320-gpu_3.0.0_all.deb",
  "./findface-data-jackfruit-160-cpu_3.0.0_all.deb",
  "./findface-data-jackfruit-160-gpu_3.0.0_all.deb",
  "./findface-data-jackfruit-320-cpu_3.0.0_all.deb",
  "./findface-data-jackfruit-320-gpu_3.0.0_all.deb",
  "./findface-data-jackfruit-480-cpu_3.0.0_all.deb",
  "./findface-data-jackfruit-480-gpu_3.0.0_all.deb",
  "./findface-data-liveness.alleyn.v2-cpu_3.0.0_all.deb",
  "./findface-data-liveness.alleyn.v2-gpu_3.0.0_all.deb",
  "./findface-data-medmask3.v2-cpu_3.0.0_all.deb",
  "./findface-data-medmask3.v2-gpu_3.0.0_all.deb",
```

(continues on next page)

(продолжение с предыдущей страницы)

```

"/findface-data-normalization-ant.v2-cpu_3.0.0_all.deb",
"/findface-data-normalization-ant.v2-gpu_3.0.0_all.deb",
"/findface-data-normalization-bee-fast-cpu_3.0.0_all.deb",
"/findface-data-normalization-bee-fast-gpu_3.0.0_all.deb",
"/findface-data-normalization-bee.v2-cpu_3.0.0_all.deb",
"/findface-data-normalization-bee.v2-gpu_3.0.0_all.deb",
"/findface-data-normalization-crop2x.v2-maxsize400-cpu_3.0.0_all.deb",
"/findface-data-normalization-crop2x.v2-maxsize400-gpu_3.0.0_all.deb",
"/findface-data-pedet-edie-rc2-cpu_3.0.0_all.deb",
"/findface-data-pedet-edie-rc2-gpu_3.0.0_all.deb",
"/findface-data-quality.v1-cpu_3.0.0_all.deb",
"/findface-data-quality.v1-gpu_3.0.0_all.deb"
],
"facen_model": "face/jackfruit_480.gpu.fnk",
"findface-extraction-api.config": {
  "listen": "127.0.0.1:18666",
  "license_ntls_server": "127.0.0.1:3133",
  "detectors": {
    "instances": 1,
    "quality_estimator": true
  },
  "normalizers": {
    "instances": 1
  },
  "extractors": {
    "instances": 1,
    "models": {
      "gender": "",
      "age": "",
      "emotions": "",
      "face": "face/jackfruit_480.gpu.fnk",
      "quality": "faceattr/quality.v1.gpu.fnk"
    }
  }
},
"findface-video-manager.config": {
  "listen": "127.0.0.1:18810",
  "master": {
    "self_url": "127.0.0.1:18811",
    "self_url_http": "127.0.0.1:18810"
  },
  "rpc": {
    "listen": "127.0.0.1:18811"
  },
  "ntls": {
    "enabled": false,
    "url": "http://127.0.0.1:3185/"
  }
},
"findface-facerouter.config": {
  "plugin_source": "dir",
  "host": "127.0.0.1",

```

(continues on next page)

(продолжение с предыдущей страницы)

```
"port": "18820",
"sfapi_url": "http://127.0.0.1:18411",
"plugin_dir": "/etc/findface-facerouter-plugins"
},
"memcached.config": {
  "max_memory": 1024,
  "item_size": 16,
  "listen_host": "127.0.0.1"
},
"findface-video-worker.config": {
  "FKVD_MGR_ADDR": "127.0.0.1:18811",
  "FKVD_NTLS_ADDR": "127.0.0.1:3133",
  "FKVD_WRK_CAP": "30",
  "streamer": [
    "port = 18999",
    "url = 127.0.0.1:18999"
  ]
},
"findface-ntls.config": {
  "NTLS_LISTEN": "127.0.0.1:3133",
  "NTLS_LISTEN_UI": "127.0.0.1:3185",
  "NTLS_LICENSE_DIR": "/opt/ntech/license"
},
"findface-tarantool-server.config": {
  "shard-001": {
    "TNT_DATA_DIR": "/opt/ntech/var/lib/tarantool/shard-001",
    "TNT_LISTEN": "127.0.0.1:32001",
    "TNT_FF_LISTEN_IP": "127.0.0.1",
    "TNT_FF_LISTEN_PORT": "8101",
    "TNT_FF_NTLS": "127.0.0.1:3133",
    "TNT_EXTRA_LUA": "\ndofile(\"/etc/findface-security/tnt_schema.lua\")\n",
    "TNT_META_SCHEME": "meta_scheme",
    "TNT_META_INDEXES": "meta_indexes"
  },
  "shard-002": {
    "TNT_DATA_DIR": "/opt/ntech/var/lib/tarantool/shard-002",
    "TNT_LISTEN": "127.0.0.1:32002",
    "TNT_FF_LISTEN_IP": "127.0.0.1",
    "TNT_FF_LISTEN_PORT": "8102",
    "TNT_FF_NTLS": "127.0.0.1:3133",
    "TNT_EXTRA_LUA": "\ndofile(\"/etc/findface-security/tnt_schema.lua\")\n",
    "TNT_META_SCHEME": "meta_scheme",
    "TNT_META_INDEXES": "meta_indexes"
  },
  "shard-003": {
    "TNT_DATA_DIR": "/opt/ntech/var/lib/tarantool/shard-003",
    "TNT_LISTEN": "127.0.0.1:32003",
    "TNT_FF_LISTEN_IP": "127.0.0.1",
    "TNT_FF_LISTEN_PORT": "8103",
    "TNT_FF_NTLS": "127.0.0.1:3133",
    "TNT_EXTRA_LUA": "\ndofile(\"/etc/findface-security/tnt_schema.lua\")\n",
    "TNT_META_SCHEME": "meta_scheme",
```

(continues on next page)

(продолжение с предыдущей страницы)

```

    "TNT_META_INDEXES": "meta_indexes"
  },
  "shard-004": {
    "TNT_DATA_DIR": "/opt/ntech/var/lib/tarantool/shard-004",
    "TNT_LISTEN": "127.0.0.1:32004",
    "TNT_FF_LISTEN_IP": "127.0.0.1",
    "TNT_FF_LISTEN_PORT": "8104",
    "TNT_FF_NTLS": "127.0.0.1:3133",
    "TNT_EXTRA_LUA": "\ndofile(\"/etc/findface-security/tnt_schema.lua\")\n",
    "TNT_META_SCHEME": "meta_scheme",
    "TNT_META_INDEXES": "meta_indexes"
  }
},
"findface-sf-api.config": {
  "listen": "127.0.0.1:18411",
  "extraction-api": {
    "extraction-api": "http://127.0.0.1:18666"
  },
  "storage-api": {
    "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": ""
      }
    ]
  },
  "limits": {
    "allow-return-facen": true
  }
},
"findface-counter.config": {
  "listen": "127.0.0.1:18300",
  "database": {
    "connection_string": "dbname=ffcounter host=/var/run/postgresql sslmode=disable"
  }
},
"pgbouncer.config": {
  "ntech_password": "hva1xogfMU03Mt6WdS4zy0XcaHs6zHOG"
},
"postgresql.config": {

```

(continues on next page)

(продолжение с предыдущей страницы)

```

    "ntech_password": "hva1xogfMU03Mt6WdS4zy0XcaHs6zHOG"
  },
  "findface-security.config": {
    "SERVICE_EXTERNAL_ADDRESS": "http://172.20.77.19",
    "FFSECURITY": {
      "ROUTER_URL": "http://127.0.0.1"
    },
    "ntech_password": "hva1xogfMU03Mt6WdS4zy0XcaHs6zHOG"
  }
}

```

1.7.3 Модели нейронных сетей

В этом разделе вы найдете сводную информацию по моделям нейронных сетей, созданным в нашей лаборатории и используемым в ПК FindFace:

Важно: При чистой установке биометрической моделью по умолчанию является jackfruit_480.

Модель	Тип нейронной сети
face/jackfruit_160	Биометрия лица
face/jackfruit_320	
face/jackfruit_480	
face/ifruit_320	
faceattr/age.v1	Распознавание возраста
faceattr/beard.v0	Распознавание бороды
faceattr/emotions.v1	Распознавание эмоций
faceattr/gender.v2	Распознавание пола
faceattr/glasses3.v0	Распознавание очков
faceattr/medmask3.v2	Обнаружение медицинской маски
faceattr/liveness.alleyн.v2	Автономный сервис Liveness
faceattr/quality.v1	Оценка качества лица

Примечание: Конфигурация для теста производительности на CPU и GPU:

- CPU - Intel® Core™ i7-6700 CPU @ 3.4GHz 4 ядра
- GPU - GeForce GTX 1080 Ti

Результаты тестирования для моделей с ускорением на CPU:

Примечание: Features в результатах теста: faceattr/age.v1.cpu, faceattr/beard.v0.cpu, faceattr/emotions.v1.cpu, faceattr/gender.v2.cpu, faceattr/glasses3.v0.cpu.

Кол-во экземпляров	1	2	3	4	5
Модель	RAM, M6				
face/ifruit_320.cpu	4300	5020	5698	7606	8279
face/ifruit_320.cpu + features	6254	7019	7725	8597	9220
face/ifruit_320.cpu + features + edie_rc2.cpu.fnk	6513	7454	9165	9685	10777
features	3034	3799	4460	5091	5774
features + medmask3.v2.cpu	3191	3942	4699	5313	6003
face/ifruit_320.cpu + medmask3.v2.cpu	4447	5108	5809	7642	8262
face/ifruit_320.cpu + edie_rc2.cpu.fnk	4583	5557	7746	8721	9592

Результаты тестирования для моделей с ускорением на GPU:

Примечание: Features в результатах теста: faceattr/age.v1.gpu.fnk, faceattr/beard.v0.gpu.fnk, faceattr/emotions.v1.gpu.fnk, faceattr/gender.v2.gpu.fnk, faceattr/glasses3.v0.gpu.fnk.

Модель	GPU RAM, M6
face/ifruit_320.gpu.fnk	3169
face/ifruit_320.gpu.fnk + edie_rc2.gpu.fnk	4243
face/ifruit_320.gpu.fnk + features	3901
face/ifruit_320.gpu.fnk + features + edie_rc2.gpu.fnk	5102
face/ifruit_320.gpu.fnk + faceattr/medmask3.v2.gpu.fnk	3313
features	2345
features + faceattr/medmask3.v2.gpu.fnk	2617

1.7.4 Хранилища данных ПК FindFace

В этом разделе:

- *Список хранилищ*
- *Галереи биометрической базы данных*

Список хранилищ

ПК FindFace использует следующие хранилища данных:

- Биометрическая база данных на основе Tarantool, в которой хранятся биометрические образцы (векторы признаков) и события идентификации лиц.
- 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 и артефакты событий (полные кадры, миниатюры лиц и нормализованные изображения лиц).
- Каталог `/var/lib/ffupload/`, в котором хранятся только такие артефакты событий, как миниатюры лиц.

Галереи биометрической базы данных

В биометрической базе данных на основе Tarantool есть 3 галереи:

- `ffsec_dossier_face`: биометрические образцы, извлеченные из фотографий в досье.
- `ffsec_events`: биометрические образцы, извлеченные из лиц, обнаруженных на видео.
- `ffsec_persons`: центроиды персон (виртуальные биометрические образцы, усредненные по всем лицам персоны) и метаданные.

1.7.5 Опции резервного копирования базы данных

Для резервного копирования биометрической базы данных 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")
```

(continues on next page)

(продолжение с предыдущей страницы)

```

-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
    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_

```

(continues on next page)

(продолжение с предыдущей страницы)

```

↪(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)

```

1.7.6 Опции восстановления базы данных

Для восстановления биометрической базы данных 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")

```

(continues on next page)

(продолжение с предыдущей страницы)

```

-cache-redis-db int
    Database to be selected after connecting to the server.
-cache-redis-network string
    Network type, either tcp or unix (default "tcp")
-cache-redis-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
    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
    
```

(continues on next page)

(продолжение с предыдущей страницы)

```
-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 ПК FindFace, включая его расширенные возможности. Данные сведения будут интересны администраторам, аналитикам, операторам и другим пользователям.

2.1 Первые шаги после установки

После успешного развертывания ПК FindFace пришло время открыть *веб-интерфейс* и начать работу. В этой главе вы найдете рекомендуемую последовательность шагов, следуя которой вы сможете задействовать весь спектр функциональных возможностей ПК FindFace.

В этой главе:

- *Подготовка к работе*
- *Создание пользователей и обеспечение безопасности системы*
- *Добавление камер*
- *Составление списков наблюдения и загрузка досье*
- *Мониторинг лиц*
- *Организация видеонаблюдения*
- *Подсчет лиц и силуэтов*
- *Анализ людей*
- *Работа в штатном режиме*
- *Базовое обслуживание системы*
- *Больше возможностей*

2.1.1 Подготовка к работе

Выполните первичную настройку системы:

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

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 ПК FindFace, 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 системный объект, который 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 дедубликации 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 системный объект, с помощью которого выполняется классификация людей по произвольным критериям: черный список, 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 лиц

По умолчанию ПК FindFace отслеживает только незарегистрированные в системе лица, т. е. лица, для которых *отсутствуют совпадения с досье*. Для того чтобы включить отслеживание лиц из пользовательского списка наблюдения, сделайте этот список *активным*. Вы также можете включить звуковое оповещение и сделать обязательным принятие вручную связанных со списком событий.

Дополнительно:

1. Добавьте информации в события, включив распознавание пола, возраста, эмоций, бороды, медицинской маски и очков. *Подробнее*.
2. Защитите систему от спуфинга, включив функцию распознавания живых лиц (Liveness). *Подробнее*.
3. 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

ПК FindFace 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 лица*, проверяя их в реальном времени на совпадение с лицами из списков наблюдения. Работайте с историей событий, используя различные фильтры.
2. Работайте с *эпизодами*. 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. Выполняйте идентификацию (поиск) лиц по следующим базам данных:
 - База данных обнаруженных лиц. *Подробнее.*
 - База данных досье. *Подробнее.*
4. *Ищите в архивных видео* лица из списков наблюдения.
5. *Сравнивайте лица* вручную, чтобы проверить их на принадлежность одному человеку.
6. *Стройте* детализированные отчеты на события распознавания лиц, эпизоды, события поиска, персоны, камеры, досье и аналитику KYC.
7. Используйте *мобильное приложение*.

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. *Используйте* журнал действий пользователей с возможностью поиска.

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 ПК FindFace 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 ПК FindFace через *HTTP API*.
3. Ознакомьтесь со списком наших *партнерских интеграций*.
4. Задействуйте *плагины*, что задать собственные правила обработки обнаруженных на видео лиц.

См.также:

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

2.2 Работа с ПК FindFace

Работа с ПК FindFace 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* ПК FindFace.

Важно: Для первого входа в систему после развертывания ПК FindFace используйте учетную запись администратора, созданную при *установке*. Для создания других пользователей см. раздел *Управление пользователями*.

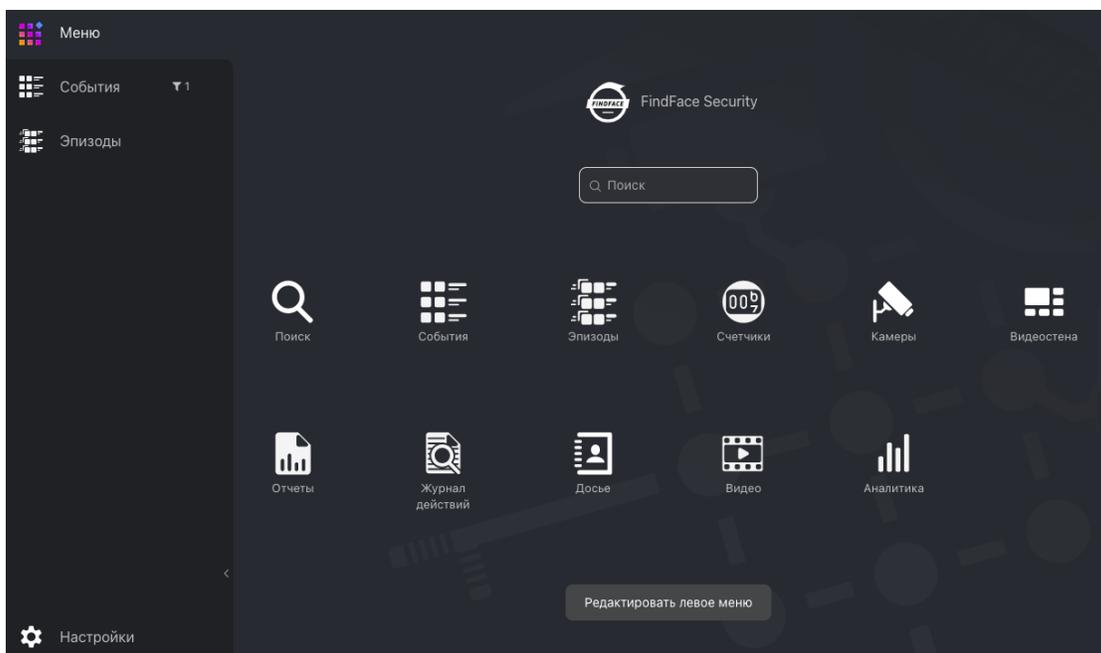
Совет: Для работы с ПК FindFace на ходу используйте *мобильное приложение*.

2.2.1 Первичная настройка

Этот раздел посвящен первичной настройке ПК FindFace. Узнайте, как настроить левую панель навигации, пороги распознавания лиц, 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 ПК FindFace используйте *Меню*.



С помощью *Меню* вы также сможете сделать ваши любимые вкладки постоянно доступными из панели навигации. Выполните следующие действия:

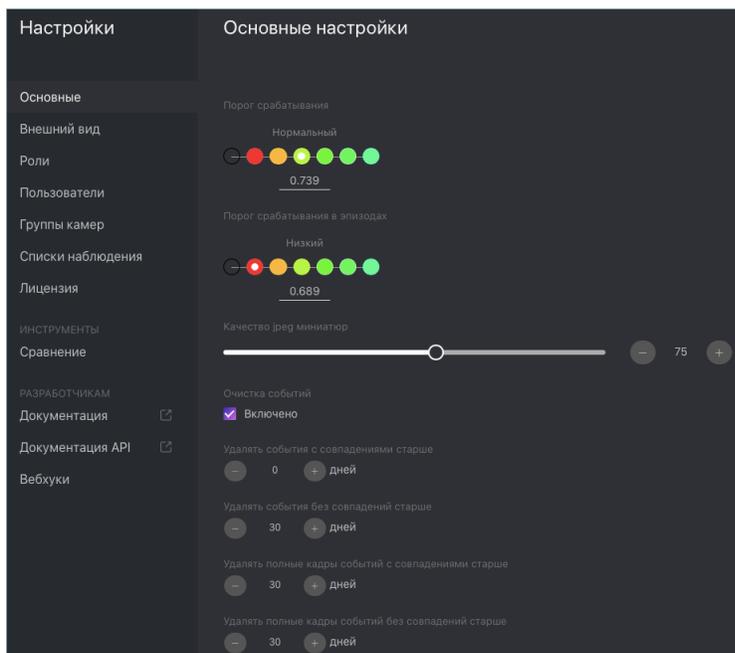
1. Нажмите *Редактировать левое меню*.
2. Установите/снимите флажки с нужных вкладок.
3. Нажмите *Завершить редактирование*.

Основные настройки

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

- универсальный порог срабатывания для распознавания лиц
- порог срабатывания для эпизодов
- 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*
- *Качество JPEG миниатюр*
- *Автоматическое удаление старых событий и эпизодов*

Универсальный порог срабатывания

ПК FindFace 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, равное 0.745. При необходимости вы можете изменить данное значение.

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

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

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

Порог срабатывания для эпизодов

При формировании эпизода система ищет в биометрической базе данных *недавние* события с похожими лицами выше определенной пороговой степени схожести. Порог по умолчанию установлен на 0.689. При необходимости вы можете изменить это значение. Обязательно свяжитесь с нашими экспертами для консультации (support@ntechlab.com).

Качество JPEG миниатюр

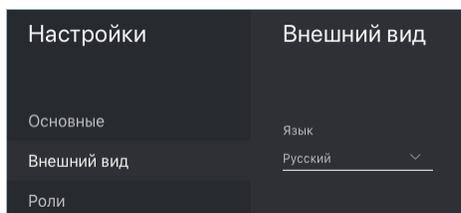
В зависимости от качества JPEG, файлы миниатюр могут в конечном счете занять значительную часть дискового пространства. Используйте вкладку *Основные* для настройки параметра.

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 досъе, а также выбрать удаление только полных кадров.

Переключение языка

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



2.2.2 Управление пользователями и безопасность системы

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

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 ПК FindFace предусмотрены следующие предустановленные роли:

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

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

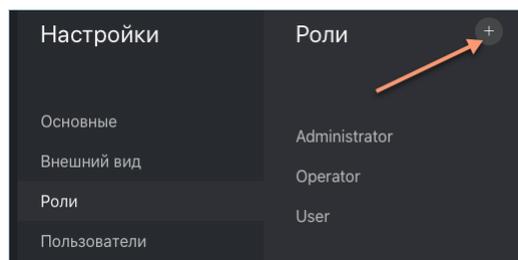
- Оператор может создавать досье вручную, получать и подтверждать события, выполнять поиск лиц в списке событий. Остальные данные доступны в режиме чтения. Возможность *пакетного создания досье* недоступна.
- Пользователь. Обладает правами только на подтверждение событий и поиск лиц в базе событий и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

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

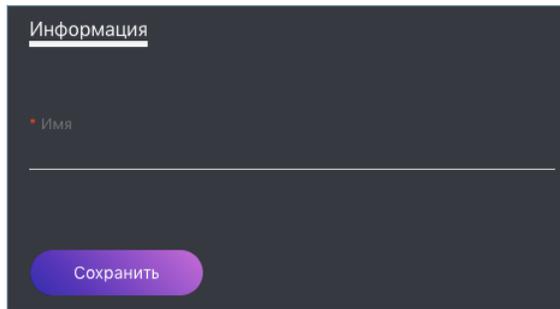
Создание новой роли

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 объектами (*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 объектов ПК FindFace:

- `dossierlist`: *списки наблюдения*;
- `dossier`: *досье*;
- `dossierface`: *фото в досье*;
- `cameragroup`: *группа камер*;
- `camera`: *камера*;
- `listevent`: *список событий*;
- `eventepisode`: *эпизоды*
- `person`: *галерея персон*
- `uploadlist`: список фотографий в *пакетной загрузке*;
- `upload`: элемент (фото) в *пакетной загрузке*;
- `user`: *пользователь*;
- `webhook`: *вебхук*;
- `videoarchive`: *идентификация лиц в видеофайлах*.
- `counter`: *счетчики лиц и силуэтов*

Вы также можете включать и отключать права на следующий функционал:

- `configure_ntls`: конфигурация *сервера лицензий findface-ntls*.
- `batchupload_dossier`: *пакетное создание досье*;

- `view_runtimesetting`: просмотр *основных настроек* ПК FindFace;
- `change_runtimesetting`: изменение основных настроек ПК FindFace
- `view_auditlog`: просмотр и работа с *аудит логами*.

Выбрать все		Отменить все						
Имя	Просмотр	Изменить	Добавить	Удалить				
dossierlist	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				
dossier	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				
dossierface	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				
cameragroup	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				
camera	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				
listevent	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				
eventepisode	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				
person	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				
uploadlist	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				
upload	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				
user	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				
webhook	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				
videoarchive	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				
counter	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				
metadictionary	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				
notification	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>						
Имя					Активный			
configure_ntls					<input checked="" type="checkbox"/>			
batchupload_dossier					<input checked="" type="checkbox"/>			
view_runtimesetting					<input checked="" type="checkbox"/>			
change_runtimesetting					<input checked="" type="checkbox"/>			
view_auditlog					<input checked="" type="checkbox"/>	4.4.999.2106+22.06b59e0 2021-0		

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.

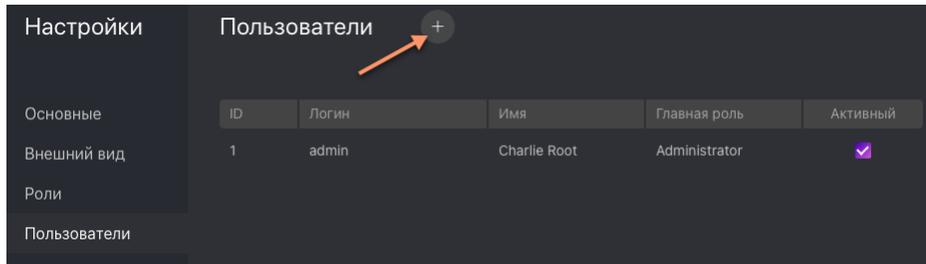
См.также:

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
User

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 ПК FindFace щелкните по его логину в списке. Нажмите *Удалить*.

Предоставление администратору прав на системные плагины

Пакет ПК FindFace 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 ПК FindFace 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 ПК FindFace возможны следующие типы аутентификации:

- `password`: стандартная аутентификация посредством логина и пароля.
- `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: *Автономный сервис liveness (findface-liveness-api)* должен быть установлен для всех типов аутентификации, основанных на распознавании лиц.

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: ПК FindFace 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 раздел `FFSECURITY_AUTH_CONFIG`.

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

(continues on next page)

(продолжение с предыдущей страницы)

```
# - FindFace Security authorization configuration dictionary -

FFSECURITY_AUTH_CONFIG = {
    # available options: face, password, face_and_password, face_or_password
    'AUTH_TYPE': 'face_or_password',
    'FACE_AUTH_CONFIDENCE': 0.745,
    # 180 days by default
    'MAXIMUM_SESSION_LENGTH': 15552000,
    # 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. Задайте параметры аутентификации:

- **AUTH_TYPE**: тип аутентификации. Доступные варианты: **face**, **password**, **face_and_password**, **face_or_password**.
- **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.
- **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. Задайте параметры мониторинга сессии пользователя:

- **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 основного сервера ПК FindFace:

```
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;  
    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;

    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-
```

(continues on next page)

(продолжение с предыдущей страницы)

```
↪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' '*';
    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;
```

(continues on next page)

(продолжение с предыдущей страницы)

```

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 ПК FindFace 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 ПК FindFace 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*.

Пользователь	IP	ID устройства	Действие	Объект
1	172.20.78.74	a430d956-3617-436d-bd8e-618f9f2ec0d1	basic_auth	user
1	172.20.78.74	85b4239a-7587-4171-a595-2997dfbd5a86	basic_auth	user
1	172.20.78.30	89d42f79-300e-4602-bc18-51d888491a7b	create	report
1	172.20.78.30	89d42f79-300e-4602-bc18-51d888491a7b	create	report
1	172.20.78.30	89d42f79-300e-4602-bc18-51d888491a7b	create	report
1	172.20.78.30	89d42f79-300e-4602-bc18-51d888491a7b	create	report
1	172.20.78.30	89d42f79-300e-4602-bc18-51d888491a7b	create	report
1	172.20.78.30	89d42f79-300e-4602-bc18-51d888491a7b	create	report
1	172.20.78.30	89d42f79-300e-4602-bc18-51d888491a7b	create	report
1	172.20.78.30	89d42f79-300e-4602-bc18-51d888491a7b	create	report

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 видео-идентификации лиц добавьте камеры в ПК FindFace, сгруппировав их с учетом расположения.

Примечание: Права на создание групп камер и камер настраиваются в разрешениях пользователя (см. *Управление пользователями*).

В этой главе:

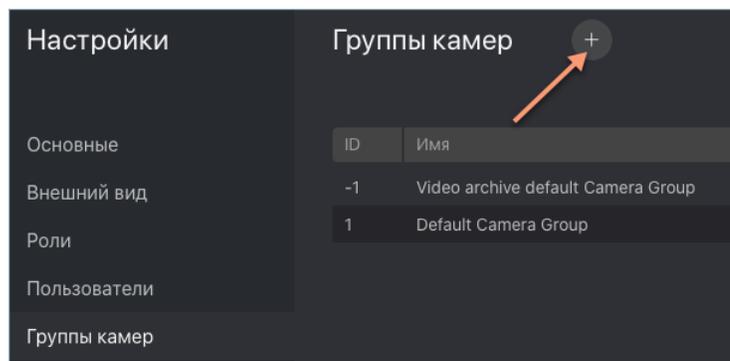
- *Создание группы камер*
- *Добавление камеры*
- *Мониторинг работы камер*

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.

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`*.

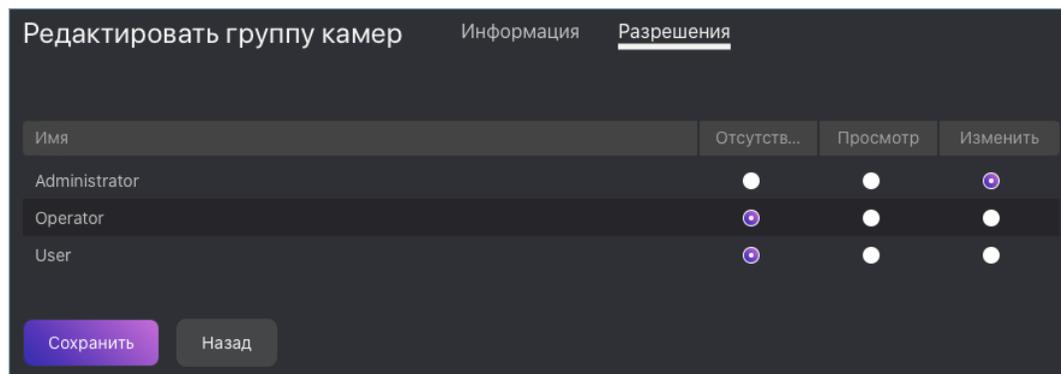
5. Если события от камер, принадлежащих одной группе, требуется дедуплицировать, т. е. исключить одинаковые события, поставьте флажок *Дедуплицировать события* и задайте в секундах интервал дедубликации (интервал, с которым события проверяются на уникальность).

Предупреждение: Используйте дедубликацию очень осторожно. Если камеры из одной группы наблюдают разные сцены, некоторые лица могут быть пропущены. Подробнее см. *Дедубликация 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.

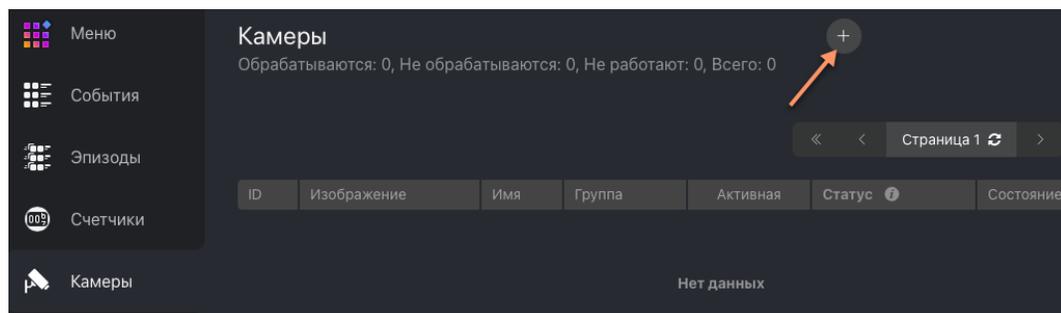


- Нажмите *Сохранить*.

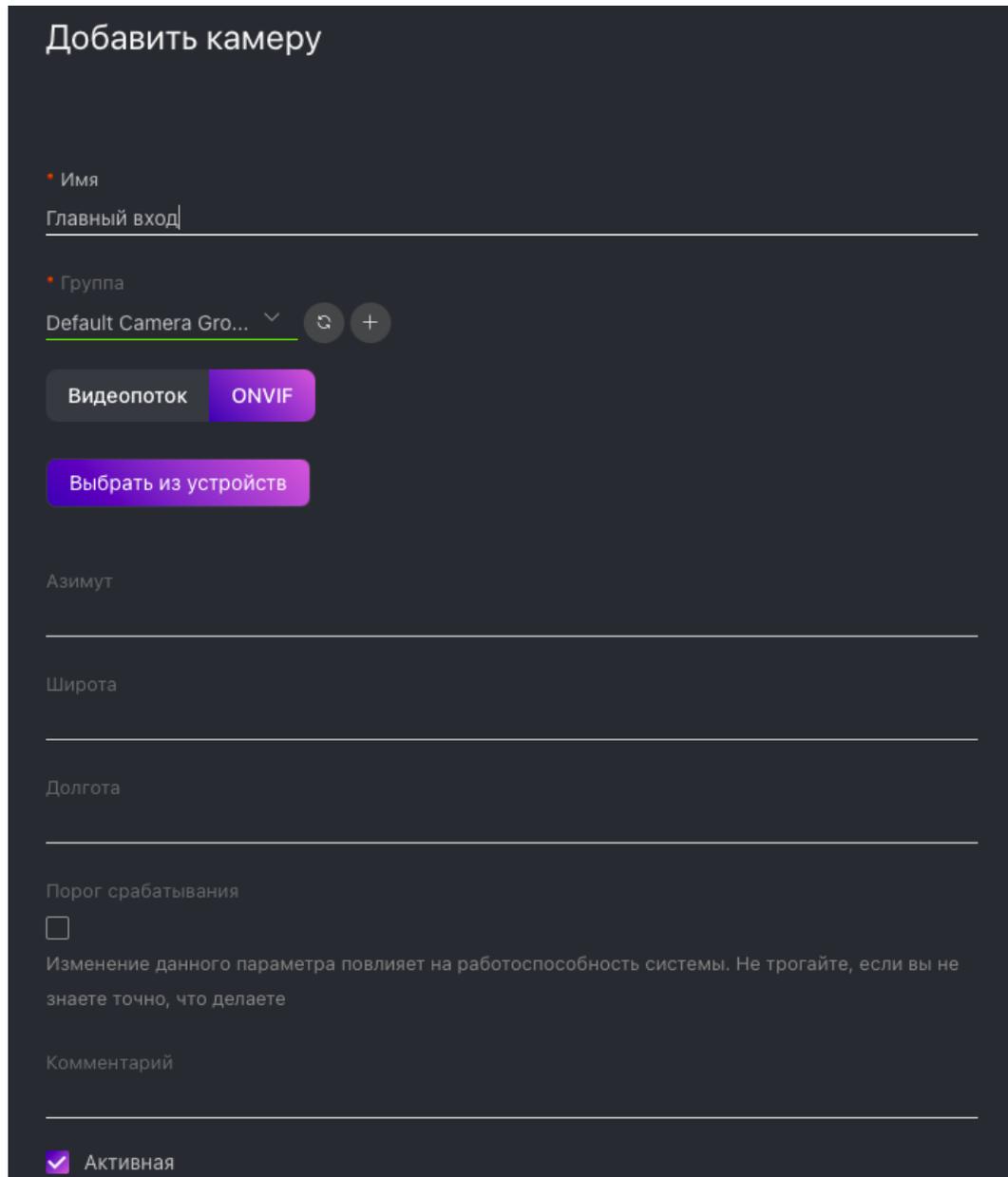
Добавление камеры

Для добавления камеры в группу выполните следующие действия:

- Перейдите на вкладку *Камеры*.
- Нажмите *+*.



- Введите название камеры и добавьте ее в одну из групп. При необходимости введите комментарий к камере.



Добавить камеру

• Имя
Главный вход

• Группа
Default Camera Gro...  

Видеопоток ONVIF

Выбрать из устройств

Азимут

Широта

Долгота

Порог срабатывания

Изменение данного параметра повлияет на работоспособность системы. Не трогайте, если вы не знаете точно, что делаете

Комментарий

Активная

4. Укажите URL камеры (*Видеопоток*). Если вы используете камеру ONVIF, выберите ее из списка обнаруженных устройств для автоматической загрузки доступных настроек и видеопотоков.
5. (Опционально) Укажите географические координаты камеры.
6. По умолчанию на всех камерах в системе используется *универсальный порог* срабатывания. Для того чтобы установить индивидуальный порог для камеры, 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.

7. Поставьте флажок *Активная*.
8. Для того чтобы настроить обработку видео, нажмите на кнопку *Параметры* и внесите изменения:
 - *Минимальное качество изображения лица (filter_min_quality)*: Минимальное качество изображения лица для отправки на сервер. Не меняйте значение по умолчанию (0.45) без предварительной консультации с нашими техническими специалистами (support@ntechlab.com).
 - *Минимальный размер лица (filter_max_face_size)*: Минимальный размер лица в пикселях для отправки на сервер.
 - *Максимальный размер лица (filter_max_face_size)*: Максимальный размер лица в пикселях для отправки на сервер.
 - *Качество сжатия (jpeg_quality)*: Качество сжатия полного кадра для отправки.
 - *Опции FFmpeg (ffmpeg_params)*: Опции ffmpeg для видеопотока. Задаются массивом строк ключ=значение, например, "rtsp_transport=tcp ss=00:20:00".
 - *Буферный режим (overall)*: Буферный режим. Отправлять для лица один кадр наилучшего качества.
 - *Временной интервал (realtime_post_interval)*: Временной интервал в миллисекундах, в течение которого в режиме реального времени выбирается лучший кадр с лицом.
 - *Отправлять первое лицо сразу (realtime_post_first_immediately)*: Если true, отправлять первое лицо из трека сразу после того, как оно пройдет через фильтры качества, размера и региона захвата (ROI), не дожидаясь окончания первого `realtime_post_interval` в режиме реального времени. Способ отправки последующих изображений лица зависит от установленного значения `realtime_post_every_interval`. Если false, отправлять первое лицо после окончания первого `realtime_post_interval`.
 - *Отправлять лучший кадр (realtime_post_every_interval)*: Если true, отправлять лучший кадр в каждом интервале времени `realtime_post_interval` в режиме реального времени. Если false, отправлять лучший кадр, только если его качество улучшилось по сравнению с предыдущим отправленным кадром.
 - *Время ожидания ответа на запрос (router_timeout_ms)*: Время ожидания в миллисекундах ответа на отправленный запрос с лицом.
 - *Получать временные метки из потока (use_stream_timestamp)*: Если true, отправлять на сервер временные метки полученные из потока. Если false, отправлять текущие дату и время.
 - *Прибавлять к временным меткам (start_stream_timestamp)*: Прибавлять указанное количество секунд к временным меткам из потока.
 - *Ограничение скорости проигрывания (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.
 - *Регион слежения (ROT)*: Включает детектирование и отслеживание лиц только внутри заданной прямоугольной области. Используйте данную опцию, чтобы уменьшить нагрузку на сервер.
 - *Регион захвата лица (ROI)*: Включает отправку на сервер лиц, обнаруженных только внутри интересующей области.

Совет: Для задания ROT/ROI удобно использовать визуальный мастер. Сначала создайте

камеру без ROT/ROI, затем откройте ее для редактирования и нажмите на кнопку *Параметры*. Вы увидите визуальный мастер.

При необходимости задайте опциональные параметры обработки видео. Для это нажмите на кнопку *Дополнительные параметры*.

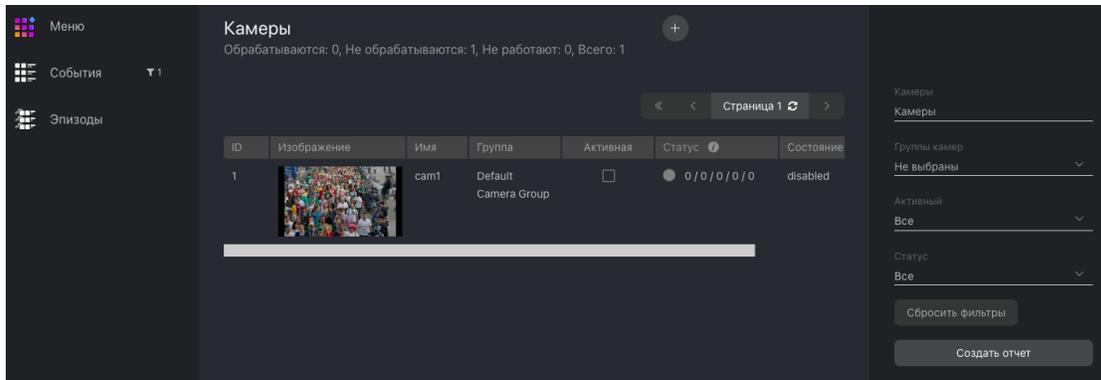
- *Формат FFMPEG (ffmpeg_format)*: Передать формат FFMPEG (mxg, flv, и т. д.), если он не может быть автоматически определен.
- *Проверять SSL-сертификат (router_verify_ssl)*: Если true, проверять SSL-сертификат сервера при отправке на него лиц через https. Если false, может быть принят самоподписанный сертификат.
- *Минимальная интенсивность движения (imotion_threshold)*: Минимальная интенсивность движения, которая будет регистрироваться детектором движения.

9. Нажмите *Сохранить*.

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 лиц.
- 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 лиц 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 faces_failed percent
    'MAX_CAMERA_FAILED_FACES': {'yellow': 0.1, 'red': 0.3},
    ...
}
```

Для каждой камеры предоставляется полная статистика: продолжительность текущей сессии обработки, количество отправленных лиц, количество лиц, обработанных с ошибками после последнего перезапуска job-задания, количество пропущенных кадров и другие данные.

Примечание: Каждая камера связана с так называемым job-задаанием, 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.*

См.также:

- *Привязка группы камер к экземпляру `findface-video-worker`*
- *Дедупликация событий*

2.2.4 Настройка мониторинга лиц. База данных досье

Данная глава посвящена настройке мониторинга лиц и созданию базы данных досье. Каждое досье должно содержать одну или несколько фотографий персоны и классифицироваться по принадлежности к тому или иному списку наблюдения, например, Розыск, VIP и т. д. Вы можете создать столько списков наблюдения, сколько необходимо.

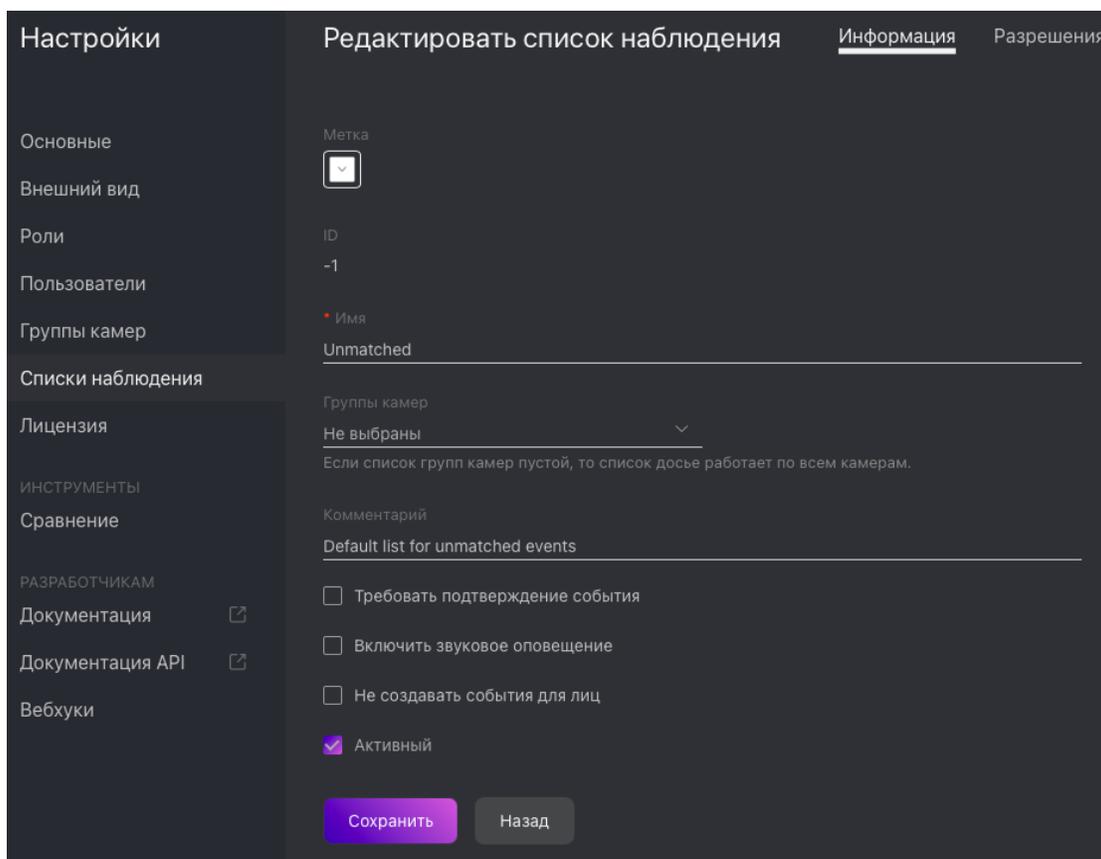
Совет: Для автоматического создания большого количества досье используйте *функционал пакетной загрузки*.

В этом разделе:

- *Мониторинг незарегистрированных лиц*
- *Создание списка наблюдения*
- *Создание досье вручную*
- *Пакетная загрузка фотографий*
- *Фильтрация досье по спискам наблюдения*
- *Очистка базы данных досье*
- *Отключение создания событий для определенных лиц*

Мониторинг незарегистрированных лиц

Базовая конфигурация ПК FindFace уже содержит предустановленный список наблюдения, предназначенный для мониторинга незарегистрированных в системе лиц, т. е. лиц, для которых отсутствуют совпадения с досье. Данный список наблюдения не может быть удален из системы. Для редактирования настроек или дезактивации списка, перейдите на вкладку *Настройки*. Выберите *Списки наблюдения* и откройте настройки списка, щелкнув *Unmatched* в таблице.



Примечание: Для того чтобы отобразить в списке событий только незарегистрированные лица,

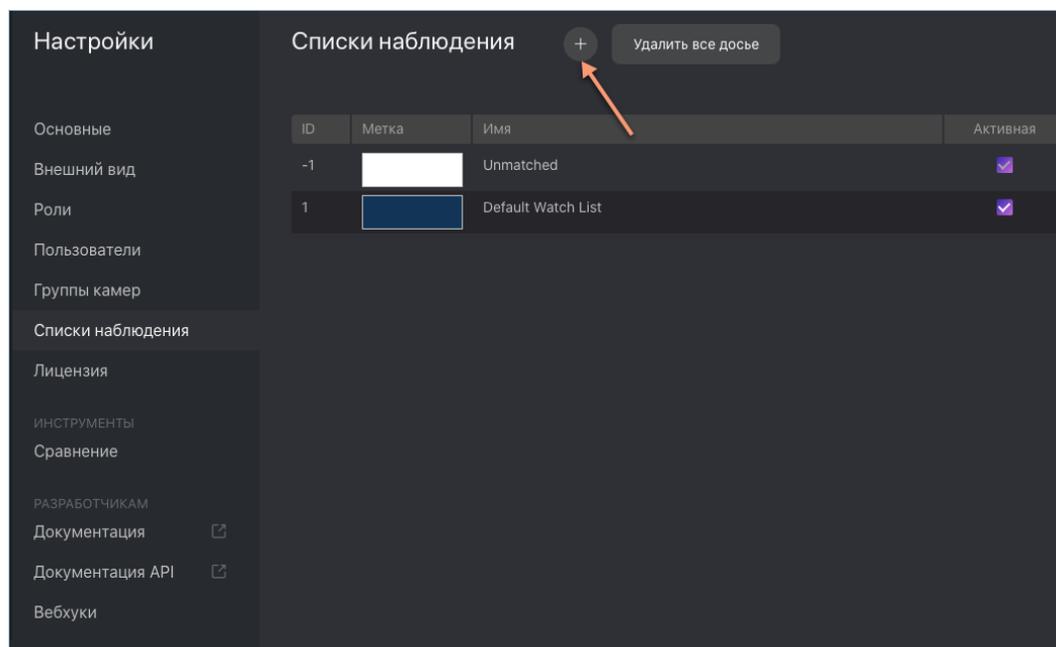
выберите значение *Unmatched* в фильтре *Списки наблюдения* на вкладке *События* (подробнее см. *Работа с событиями*).

Создание списка наблюдения

Для создания пользовательского списка наблюдения выполните следующие действия:

Совет: Помимо списка наблюдения *Unmatched* для мониторинга незарегистрированных лиц, в системе по умолчанию уже создан список для мониторинга лиц по базе досье. Данный список наблюдения не может быть удален из системы.

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

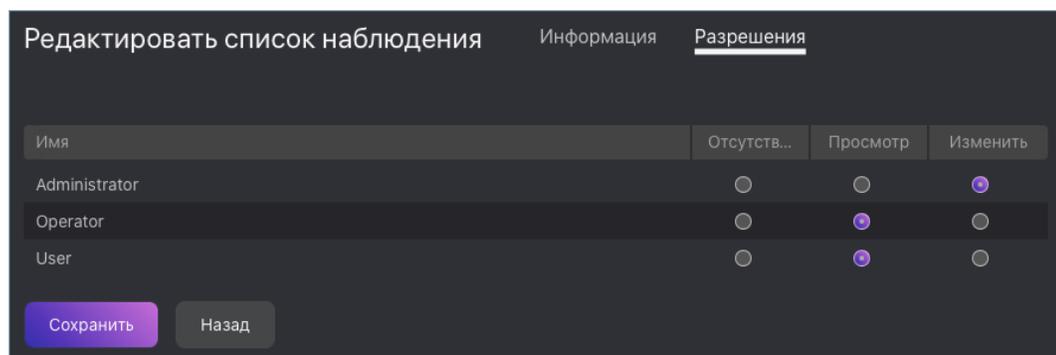


3. В палитре *Метка* выберите цвет, который будет использоваться в событиях распознавания людей из данного списка. Правильно выбранный цвет повышает быстроту реагирования оператора на событие.

4. Введите название списка наблюдения. При необходимости добавьте комментарий.
5. Выберите группу камер, которая будет использоваться для мониторинга списка наблюдения. Если группа камер не выбрана, список наблюдения будет отслеживаться всеми активными камерами в системе.
6. Поставьте флажок *Требовать подтверждение*, если для данного списка оператор должен в обязательном порядке подтвердить принятие события.
7. При необходимости включите звук при появлении события для данного списка.
8. По умолчанию ко всем спискам наблюдения в системе применяется *универсальный порог срабатывания*. Для того чтобы задать индивидуальный порог для списка наблюдения, поставьте флажок *Порог срабатывания* и укажите нужное значение.

Важно: Для большинства случаев распознавания лиц значение, установленное по умолчанию для универсального порога срабатывания, является оптимальным. Не рекомендуется самостоятельно менять его. Обязательно 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.

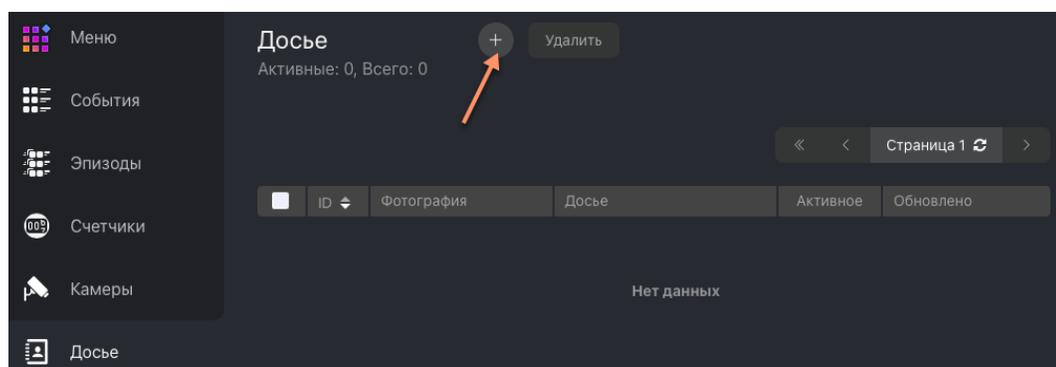


12. Нажмите *Сохранить*.

Создание досье вручную

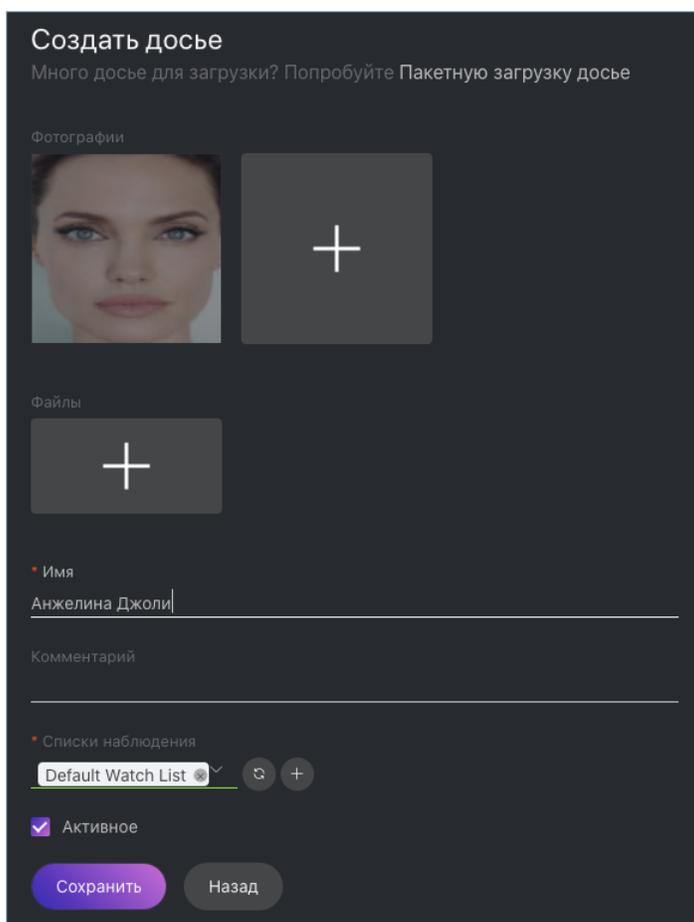
Для создания досье вручную выполните следующие действия:

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



3. Добавьте одну или несколько фотографий, имеющие отношение к делу файлы и введите имя человека. При необходимости добавьте 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.



4. Из раскрывающегося списка *Списки наблюдения* выберите список (или несколько списков, по очереди), в который следует добавить досье.
5. Поставьте флажок *Активное*. Если досье неактивно, оно не будет использоваться для идентификации лица в режиме реального времени.
6. Нажмите *Сохранить*. Если похожее досье уже существует в базе данных, вы сможете выбрать, объединить ли его с новым досье, отменить создание нового досье, в любом случае создать новое.

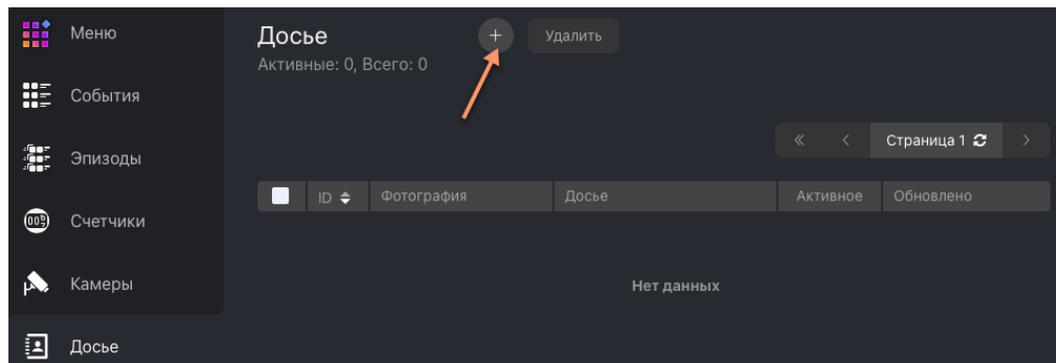
Пакетная загрузка фотографий

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

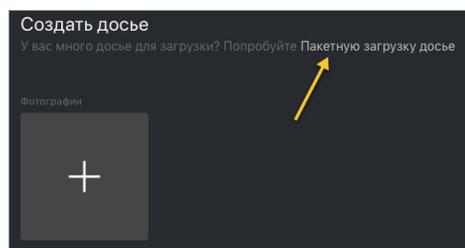
Совет: Если количество досье превышает 10000, используйте *Пакетная загрузка фотографий через консоль*.

Важно: Лицо на фотографиях 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.

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



3. Нажмите *Пакетная загрузка досье*.



4. Выберите фотографии для загрузки пофайлово или укажите папку с фотографиями.

Пакетная загрузка досье

Журналы

Выбрать файлы или Выбрать директорию

Использовать имя файла как имя

Префикс имени

Постфикс имени

Использовать имя файла как комментарий

Префикс комментария

Постфикс комментария

* Списки наблюдения

Выбрать

Параллельная загрузка

2 5 10 20

Групповая фотография

Отклонить

- 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.

- Из раскрывающегося списка *Списки наблюдения* выберите список (или несколько списков, по очереди), в который следует добавить создаваемые досье.
- В параметре *Параллельная загрузка* задайте количество потоков загрузки фотографий. Чем больше потоков, тем быстрее будет завершена загрузка, однако также потребуется и большее количество ресурсов.
- Из раскрывающегося списка *Групповая фотография* выберите, как должна поступить система при наличии нескольких лиц на фотографии: отклонить фотографию, загрузить самое большое лицо, загрузить все лица.
- 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 ↻ > >

Фильтрация досье по спискам наблюдения

Все созданные в ПК FindFace досье отображаются на вкладке *Досье*. Используйте фильтр *Списки наблюдения*, чтобы отфильтровать досье по спискам.

Досье + Удалить
Активные: 3, Всего: 3

<< < Страница 1 ↻ > >

<input type="checkbox"/>	ID	Фотография	Досье	Активное	Обновлено
<input type="checkbox"/>	3		Default Watch List	<input checked="" type="checkbox"/>	2020-12-25 21:12:23
<input type="checkbox"/>	2		Default Watch List	<input checked="" type="checkbox"/>	2020-12-25 20:26:45
<input type="checkbox"/>	1		Default Watch List	<input checked="" type="checkbox"/>	2020-12-25 20:23:56

Досье
Досье

Списки наблюдения
Не выбраны

- Default Watch List

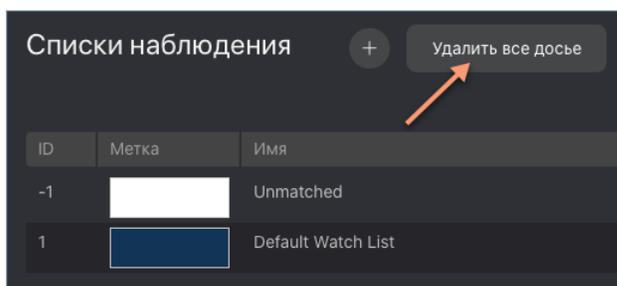
ID досье

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*.

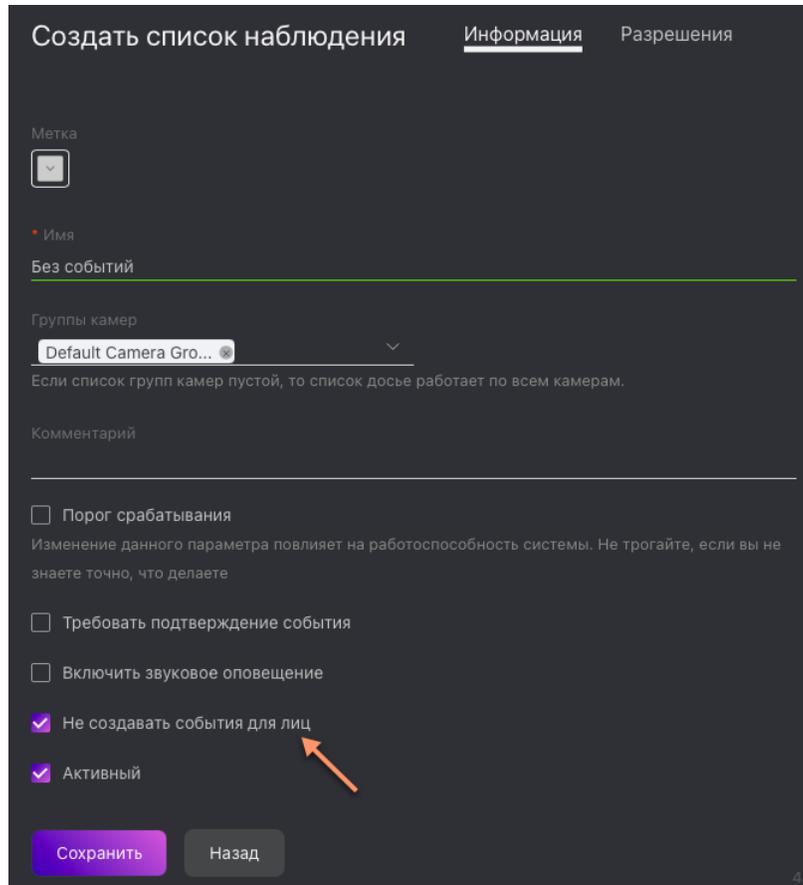


Отключение создания событий для определенных лиц

Иногда при мониторинге необходимо пропускать определенные лица. Один из самых распространенных случаев — это лица 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 лиц, не подлежащих детектированию. В его настройках установите флажок *Не создавать событий для лиц*.



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

2.2.5 Счетчики лиц и силуэтов

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

ПК FindFace 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.

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.

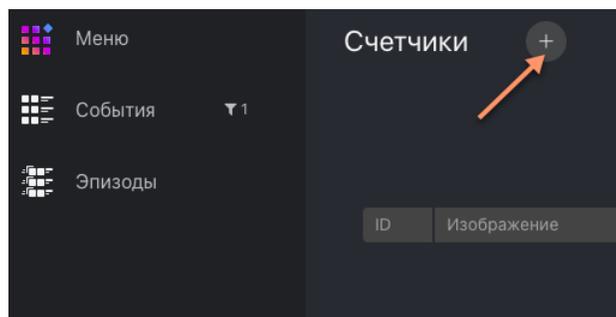
```
sudo vi /etc/findface-security/config.py

# counters full frame saving options:
# `always` - save always
# `detect` - save only if faces or silhouettes 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.

Создать счетчик Информация

* Имя
Счетчик метро

* Камера
cam1

* Интервал подсчета (секунды)
5

Определение лиц

Определение силуэтов
Чтобы использовать счётчик по силуэтам необходимо включить детектирование силуэтов. Для этого ознакомьтесь с документацией (раздел "Включение детектирования силуэтов"), или обратитесь к администратору вашей установки FindFace Security.

Активный

Сохранить Назад

4. Выберите камеру для подсчета.
5. Задайте интервал между двумя последовательными скриншотами, используемыми для подсчета лиц и силуэтов.
6. Установите флажок *Определять лица* для подсчета лиц.
7. Установите флажок *Определять силуэты* 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 *Регион захвата*, чтобы задать регион отслеживания лиц и силуэтов в поле зрения камеры.

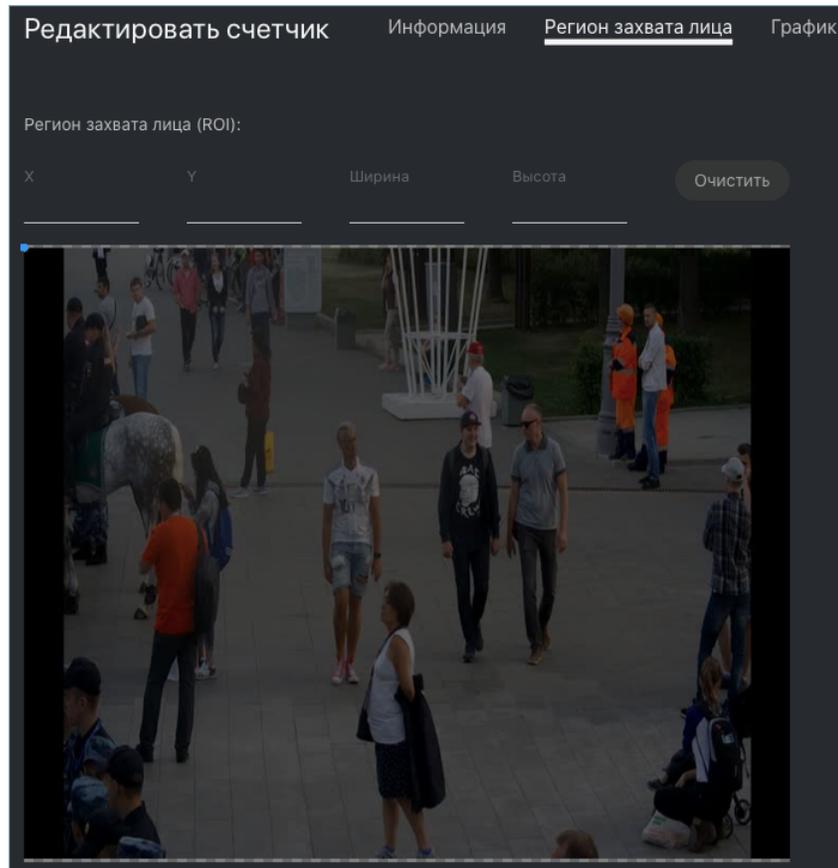
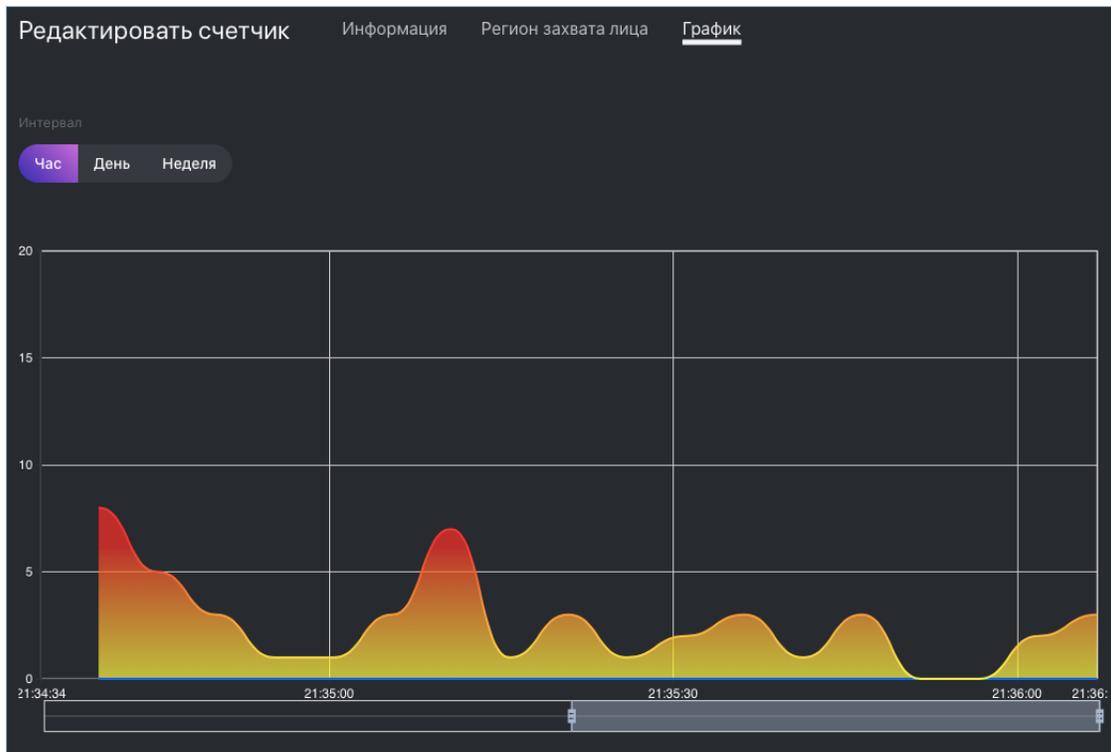


График счетчика

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* и нажмите на нужный счетчик.

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

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, если запись счетчика содержит определенное количество лиц и силуэтов.

См.также:

- *Детектирование силуэтов*
- *Вебхуки*

2.2.6 События и эпизоды распознавания лиц

Результат работы системы по части идентификации лиц на видеоизображении в режиме реального времени отображается на вкладках *События* и *Эпизоды*. Помимо работы с текущими событиями идентификации, данные вкладки также предоставляют доступ к истории событий.

Совет: Поиск лица в списке событий и базе данных досье с эталонными изображениями лиц выполняется на вкладке *Поиск*.

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

Работа с событиями

Данный раздел посвящен вкладке *События*.

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

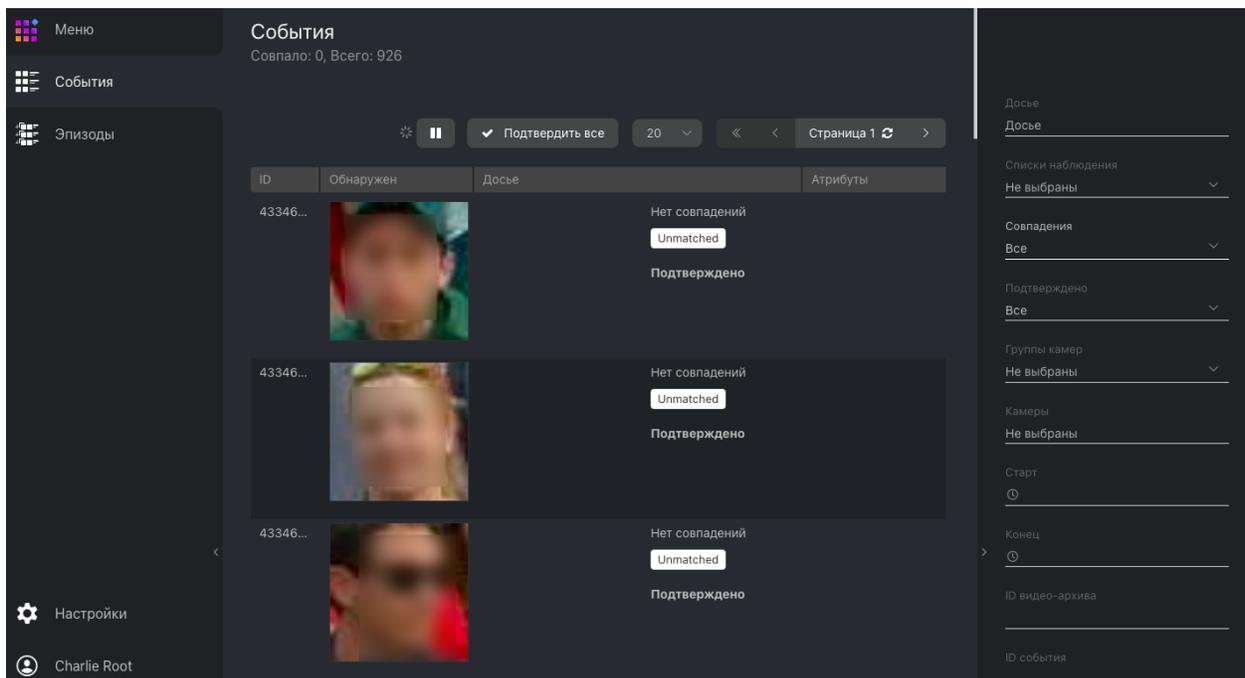
Важно: Вы можете *включить звуковые уведомления* для событий, связанных с конкретными списками наблюдения. В некоторых браузерах для того чтобы воспроизводился звук, вкладка с событиями должна оставаться в фокусе. Чтобы выделить вкладку, откройте ее и щелкните в любом месте страницы.

В этой главе:

- Просмотр событий идентификации в режиме реального времени
- Распознавание живых лиц и атрибутов лица
- Карточка события. Принятие события
- Карточка события. Поиск лица

Просмотр событий идентификации в режиме реального времени

При обнаружении лица в списке событий выводится уведомление.



Уведомление содержит следующую информацию:

- Если на лицо отсутствует досье: нормализованное изображение лица, дата и время обнаружения лица, группа камер.
- Если на лицо заведено досье: нормализованное изображение лица, фотография из досье, имя персоны, степень схожести лиц, комментарий из досье, список досье, дата и время обнаружения лица, группа камер.

Примечание: Система может быть настроена таким образом, что уведомления будут выводиться только для лиц 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:

- *Досье*: отображать только события по определенному досье.
- *Списки наблюдения*: отображать только события по определенному списку наблюдения.

Примечание: Для просмотра только лиц без совпадений в списке событий, установите в фильтре *Без совпадений*.

- *Совпадения*: отображать только события с совпадением лиц/без совпадения лиц или все события.
- *Подтверждено*: отображать только принятые/непринятые или все события.
- *Камеры*: отображать только события по определенной камере.
- *Группы камер*: отображать только события по определенной группе камер.
- *Старт, Конец*: отображать только события, случившиеся в определенный период времени.
- *ID видеоархива*: отобразить события из видеоархива с определенным ID.
- *ID события*: отобразить событие с определенным ID.
- *ID эпизода*: отобразить события из эпизода с определенным ID.

Распознавание живых лиц и атрибутов лица

В зависимости от настроек системы, вы можете видеть оценку Liveness лица, а также результат распознавания атрибутов лица, таких как пол, возраст, эмоции, наличие очков и бороды.

Детектор живых лиц (Liveness) в автоматическом режиме отличает настоящее лицо от лица на фото- или видеоизображении, предотвращая мошенничество.

Примечание: Результат Liveness может быть null. Так происходит, когда детектор живых лиц отключен или не может достоверно оценить Liveness на предоставленном изображении.

Результат распознавания атрибутов лица возвращается в следующем формате:

Атрибут лица	Формат результата	Пример
Возраст	Атрибут: возраст : число лет	возраст: 33
Пол	Результат: мужской/женский (атрибут: пол): уверенность алгоритма в результате	женский (пол): 0.95
Эмоции	Результат: злость/отвращение/страх/счастье/грусть/удивление (атрибут: эмоции): уверенность алгоритма в результате	счастье (эмоции): 0.99
Очки	Результат: медицинские/солнечные/нет (атрибут: очки): уверенность алгоритма в результате	нет (очки): 0.87
Борода	Результат: борода/нет (атрибут: борода): уверенность алгоритма в результате	нет (борода): 0.91

При необходимости выполните фильтрацию событий по атрибутам лиц, отобразите только живые лица/попытки спуфинга.

Возраст
От До

Борода
 Отсутствует Борода

Эмоции
 Злость Отвращение
 Страх Счастье
 Грусть Удивление
 Нейтральное состояние

Пол
 Мужской Женский

Очки
 Отсутствует Медицинские
 Солнечные

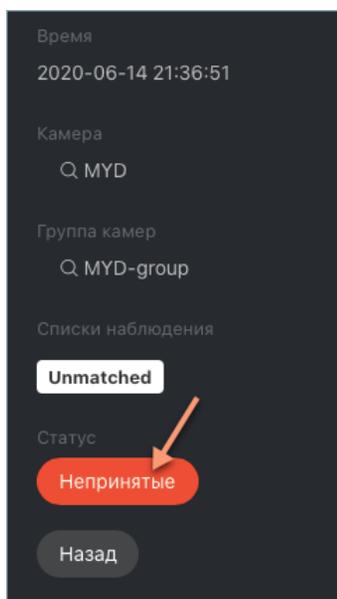
Liveness
 Живой человек
 Изображение

Медицинская маска
 Отсутствует
 Неправильно надета
 Правильно надета

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*.



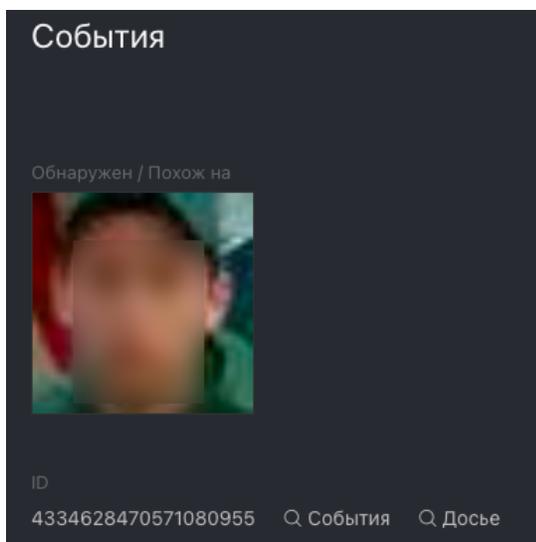
Совет: Если на обнаруженное лицо заведено досье, в него можно перейти, щелкнув по имени персоны в карточке события.

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 лица

ПК FindFace позволяет искать обнаруженные лица в базе данных обнаруженных лиц и в базе данных досье с эталонными изображениями лиц. Для перехода на вкладку поиска из карточки события нажмите *События* и *Досье*.



См.также:

- [Поиск по базам данных.](#)

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.

Совет: Поиск лица в списке событий и базе данных досье с эталонными изображениями лиц выполняется на вкладке *Поиск*.

Совет: Для идентификации лиц в архивных видео см. [Идентификация лиц в офлайн видео](#).

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 биометрической базе данных 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,
    ...
}

...
```

См.также:

Для того чтобы посмотреть, как работают эпизоды, перейдите на вкладку *Эпизоды*. Подробнее см. *Эпизоды событий*.

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`.

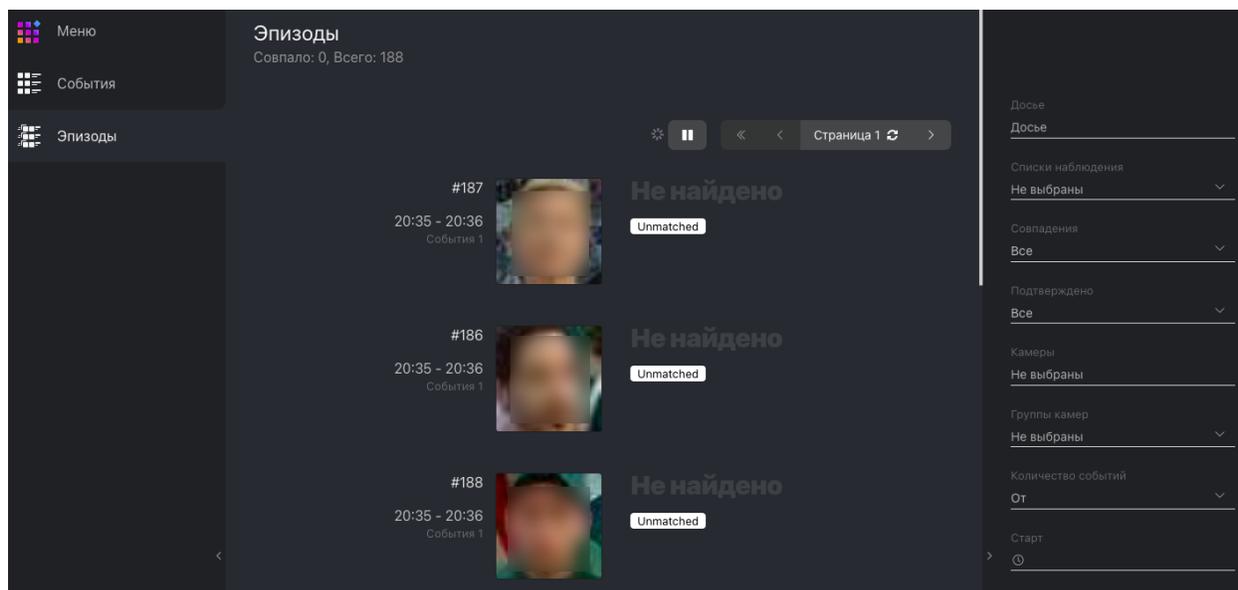
Совет: См. *Управление пользователями*.

Выбрать все		Отменить все						
Имя	Просмотр	Изменить	Добавить	Удалить				
dossierlist	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				
dossier	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				
dossierface	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				
cameragroup	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				
camera	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				
listevent	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				
eventepisode	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				
person	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				
uploadlist	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				
upload	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				
user	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				
webhook	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				
videoarchive	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				
counter	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				
metadictionary	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				
notification	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>						
Имя					Активный			
configure_ntls					<input checked="" type="checkbox"/>			
batchupload_dossier					<input checked="" type="checkbox"/>			
view_runtime-setting					<input checked="" type="checkbox"/>			
change_runtime-setting					<input checked="" type="checkbox"/>			
view_auditlog					<input checked="" type="checkbox"/>			

4.4.999.2106+22_g6b59e0 2021-0

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:

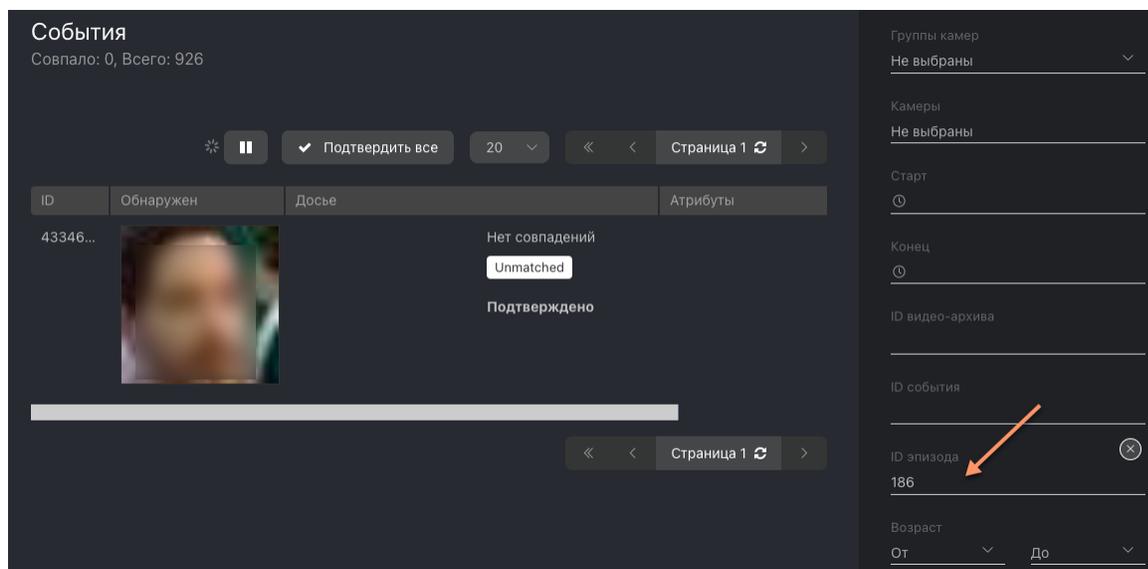
- *Досье*: отображать только эпизоды по определенному досье.
- *Списки наблюдения*: отображать только эпизоды по определенному списку наблюдения.

Примечание: Для просмотра только лиц без совпадений в списке эпизодов, установите в фильтре *Без совпадений*.

- *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).

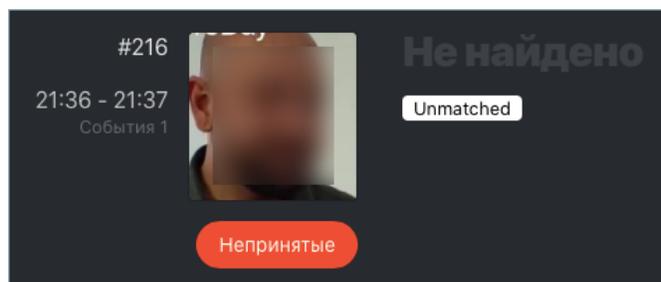
Для просмотра событий щелкните по нужному эпизоду в списке. Вы будете переправлены на вкладку *События* с соответствующим ID эпизода в фильтре *Эпизод*:



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 лиц в офлайн видео

Помимо идентификации лица в режиме реального времени, ПК FindFace также позволяет выполнять идентификацию лиц в офлайн видео. Данная функциональность имеет широкий спектр возможных применений, среди которых наиболее распространенным случаем является обнаружение и распознавание лиц 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,
    ...
}

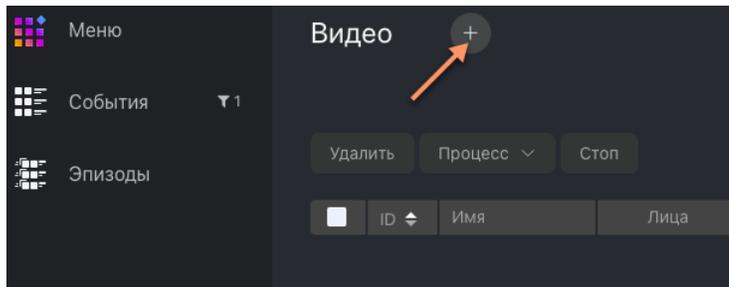
...
```

Обработка видеофайла

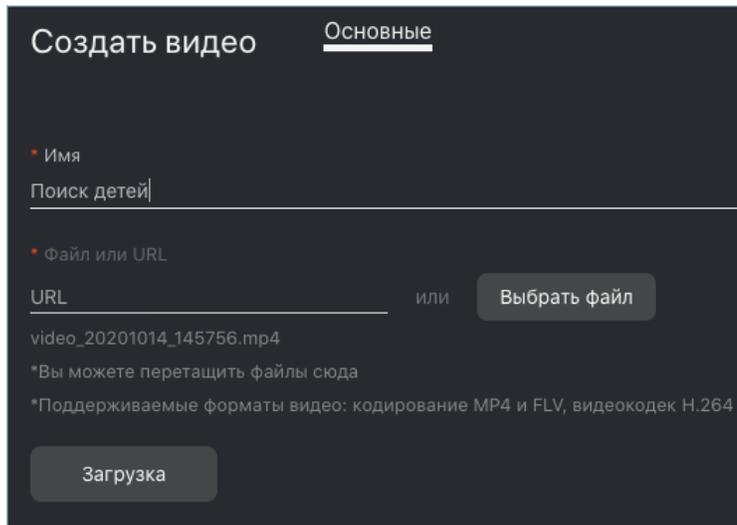
Для идентификации лица в офлайн видео выполните следующие действия:

1. Для обработки офлайн-видео нужна группа камер. Вы можете создать новую группу камер с базовыми настройками или использовать группу камер Video Archive default, уже существующую в системе и предназначенную для обработки видеоархивов. После того как группа камер выбрана, укажите ее в настройках тех *списков наблюдения*, лица в которых нужно найти на видео.
2. Создайте видео в ПК FindFace, загрузив его из файла или онлайн-хранилища/облака. Для этого, перейдите на вкладку *Видео*.

3. Нажмите +.



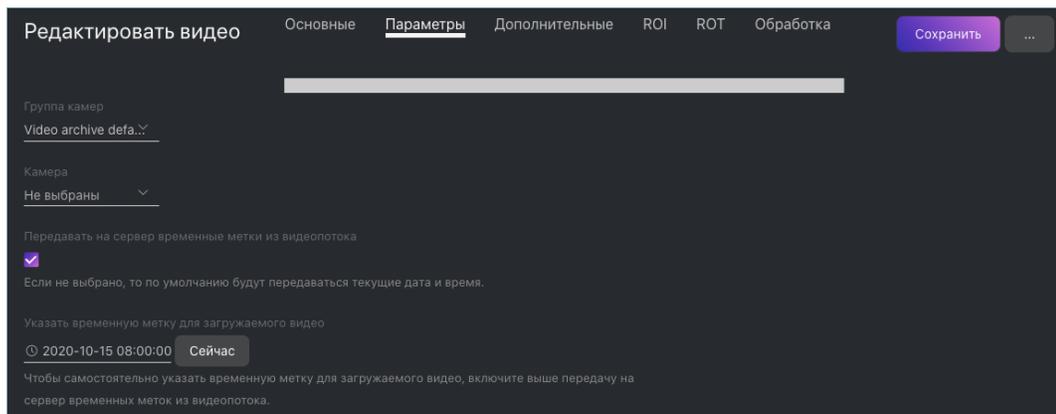
4. Укажите название видео.



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

6. Нажмите *Загрузка*.

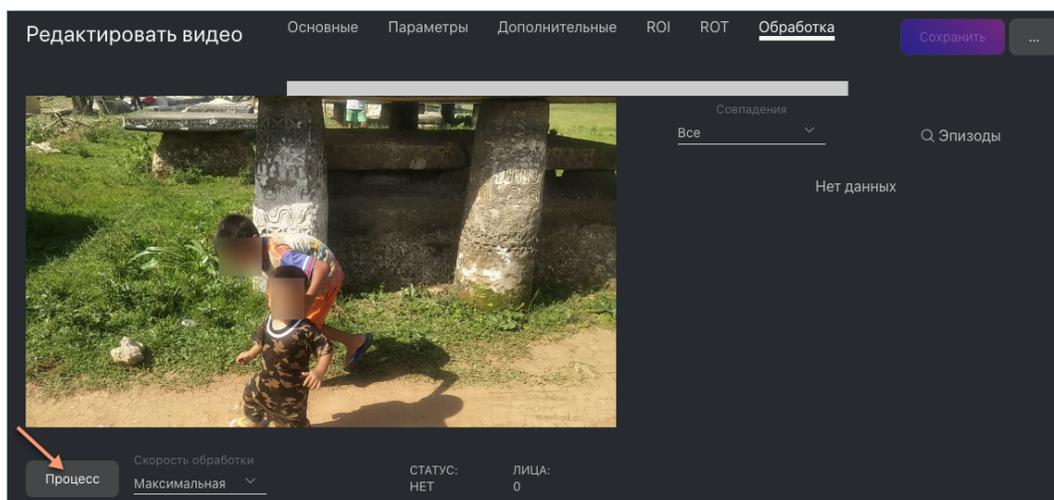
7. По завершении загрузки видео перейдите на вкладку *Параметры*. Выберите используемую группу камер.



8. (Опционально) Выберите камеру, с которой будут ассоциироваться лица, найденные на видео.

9. (Опционально) Настройте метки времени для событий распознавания лиц.

10. (Опционально) На вкладках *Дополнительные*, *ROI*, *ROT* укажите параметры обработки видео по аналогии с параметрами обработки видеопотока с *камеры*.
11. Перейдите на вкладку *Обработка*. Нажмите *Процесс*, чтобы запустить идентификацию лиц.



Вы можете просмотреть события идентификации лица прямо здесь, а также на вкладках *События* и *Эпизоды*, отфильтровав список событий по группе камер/ камере, ассоциированной с видео.

2.2.8 Поиск по базам данных

ПК FindFace позволяет выполнять идентификацию (поиск) лиц по следующим базам данных:

- База данных обнаруженных на видео лиц (вкладка *События*).
- База данных досье (вкладка *Досье*). Содержит эталонные изображения лиц.

Поиск лиц выполняется на вкладке *Поиск*.

В этой главе:

- Поиск лица в базе данных обнаруженных лиц
- Поиск лица в базе данных досье

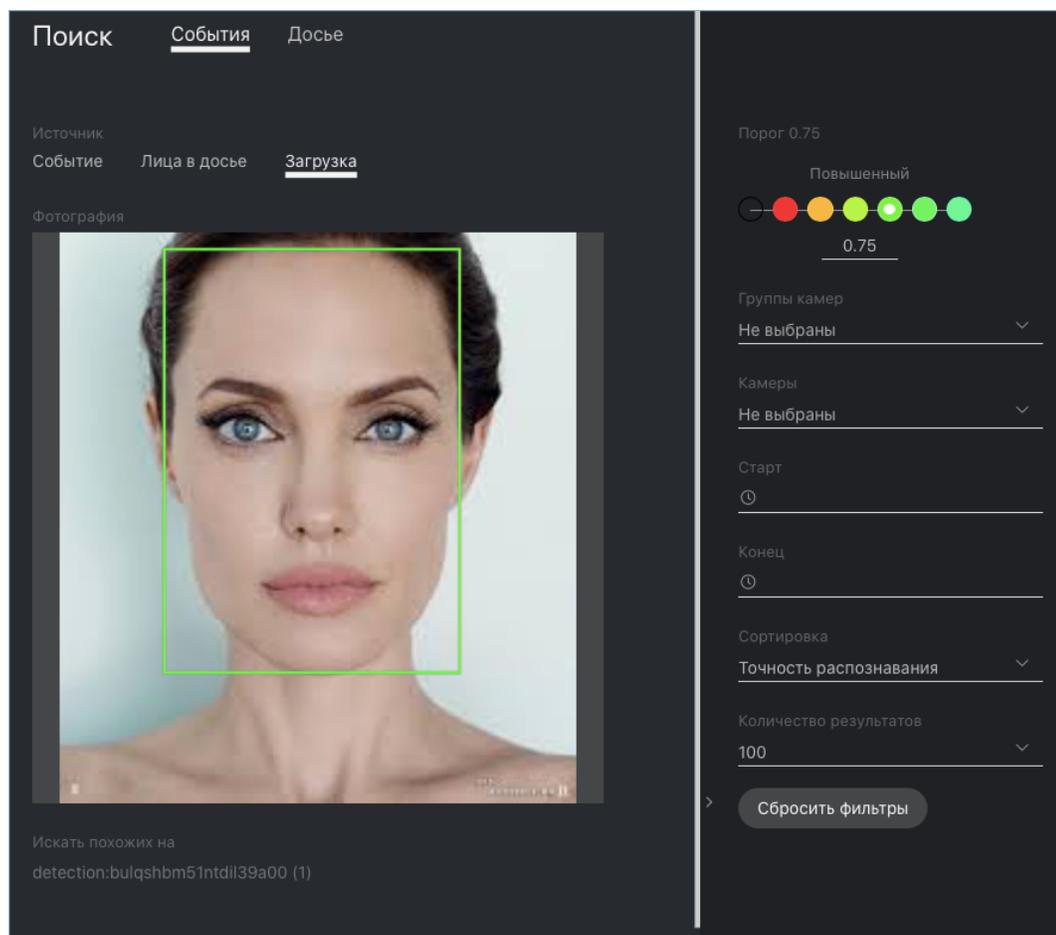
Поиск лица в базе данных обнаруженных лиц

ПК FindFace позволяет выполнять идентификацию лица по базе данных обнаруженных на видео лиц.

Примечание: В интерфейсе база данных представлена списком событий (вкладка *События*).

Для идентификации лица по базе данных выполните следующие действия:

1. Перейдите на вкладку *Поиск*.



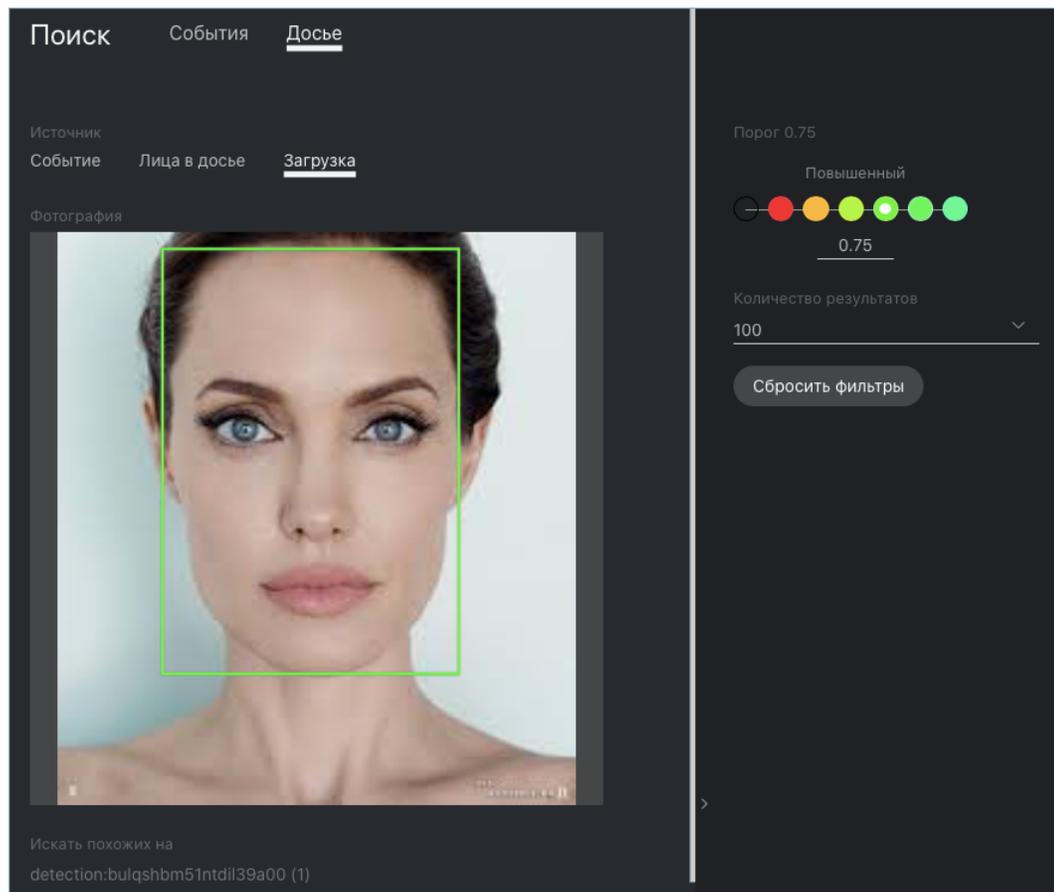
2. Укажите место поиска: *События*.
3. Задайте лицо, которое требуется найти, одним из следующих способов:
 - Задав ID события с искомым лицом.
 - Задав ID досье с искомым лицом. Если досье содержит несколько фотографий, выберите те, которые будут использованы для поиска.
 - Загрузив фотографию. Фотография будет отображена в одноименном поле. Если на фотографии присутствует несколько лиц, выберите нужное.
4. По умолчанию в результатах поиска отображаются лица, степень схожести которых с искомым равна или превышает 0.75. При необходимости измените данное значение.
5. При необходимости укажите группу камер и период времени, в течение которого произошло событие.
6. Выберите метод сортировки результатов поиска: по степени уверенности алгоритма в совпадении лиц (степени схожести между лицами) или дате.
7. Укажите максимальное количество событий в результатах поиска.
8. Нажмите *Поиск*. Результаты поиска будут отображены ниже. Для каждого найденного лица будет указана вероятность его совпадения с лицом на фотографии.

Поиск лица в базе данных досье

ПК FindFace позволяет выполнять идентификацию лица по базе данных, содержащей досье с эталонными изображениями лиц.

Для идентификации лица по базе данных выполните следующие действия:

1. Перейдите на вкладку *Поиск*.



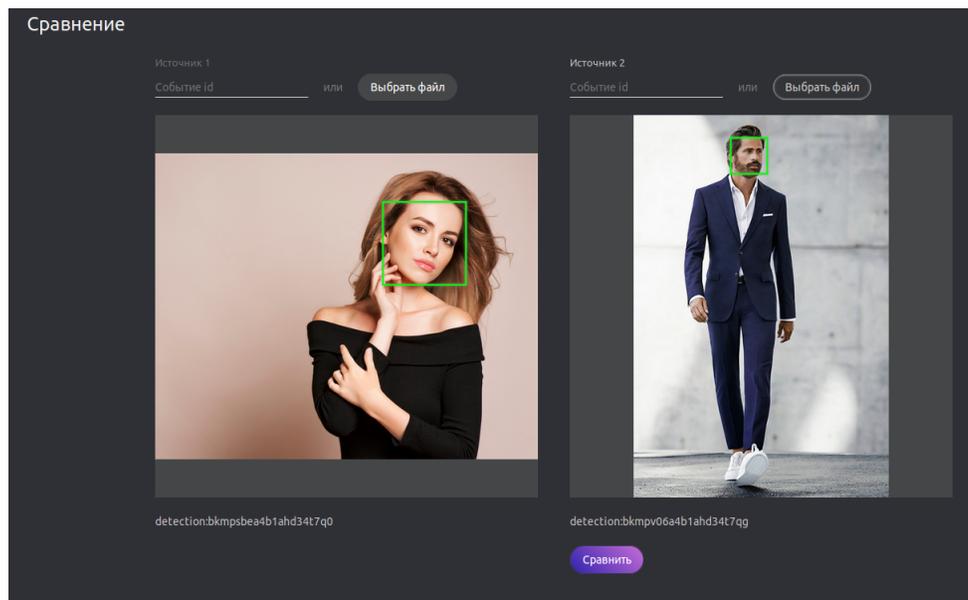
2. Укажите место поиска: *Досье*.
3. Задайте лицо, которое требуется найти, одним из следующих способов:
 - Задав ID события с искомым лицом.
 - Задав ID досье с искомым лицом. Если досье содержит несколько фотографий, выберите те, которые будут использованы для поиска.
 - Загрузив фотографию. Фотография будет отображена в одноименном поле. Если на фотографии присутствует несколько лиц, выберите нужное.
4. По умолчанию в результатах поиска отображаются лица, степень схожести которых с искомым равна или превышает 0.75. При необходимости измените данное значение.
5. Выберите метод сортировки результатов поиска: по степени уверенности алгоритма в совпадении лиц (степени схожести между лицами) или дате.
6. Укажите максимальное количество событий в результатах поиска.

7. Нажмите *Поиск*. Результаты поиска будут отображены ниже. Для каждого найденного лица будет указана вероятность его совпадения с лицом на фотографии.

2.2.9 Сравнение двух лиц

ПК FindFace позволяет выполнять сравнение 2-х лиц. Выполните следующие действия:

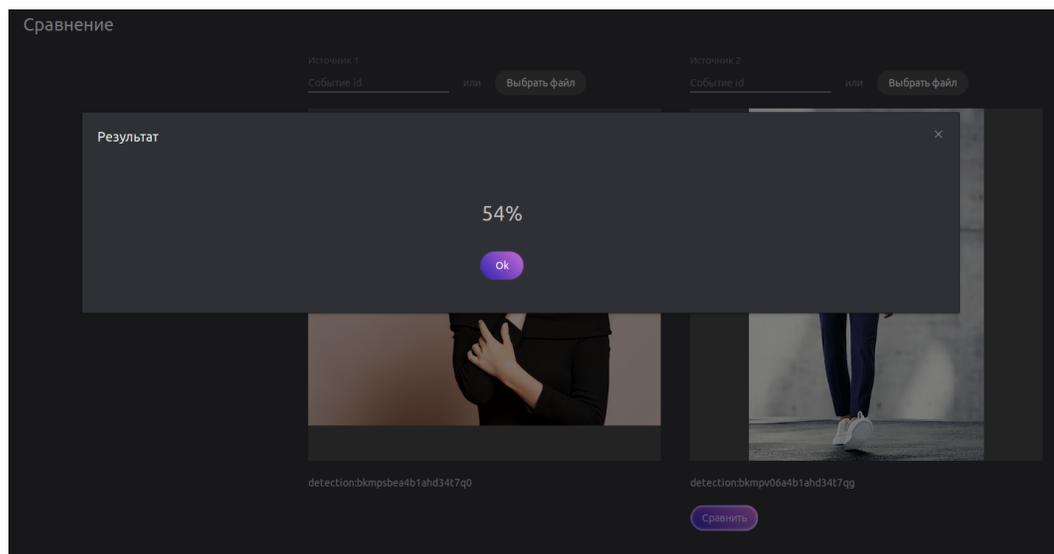
1. Перейдите на вкладку *Настройки*. Выберите *Сравнение*.



2. Укажите id событий, лица из которых нужно сравнить, и/или загрузите фотографии с лицами.

Совет: Узнать ID события можно на вкладке *События*.

3. Нажмите *Сравнить*. В результате будет отображена вероятность принадлежности лиц одному человеку.



2.2.10 Распознавание персон и связанная с людьми аналитика

ПК FindFace является идеальным инструментом для сбора связанной с людьми аналитики. Сначала включите распознавание персон, а затем извлеките из него максимум информации, используя набор аналитических функций.

Распознавание персон

ПК FindFace 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

ПК FindFace 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, биометрический образец и миниатюру лица, а также ссылку на родительский эпизод.
- Выполняет поиск похожего центроида лица в галерее `person_events` биометрической базы данных Tarantool. Центроид лица — это виртуальный биометрический образец, усредненный по всем обнаруженным лицам персоны. Если похожий центроид найден, система

обновляет его, используя биометрический образец из нового события. В противном случае создается новый центрост.

- 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 биометрических образцов.

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 ПК FindFace 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.

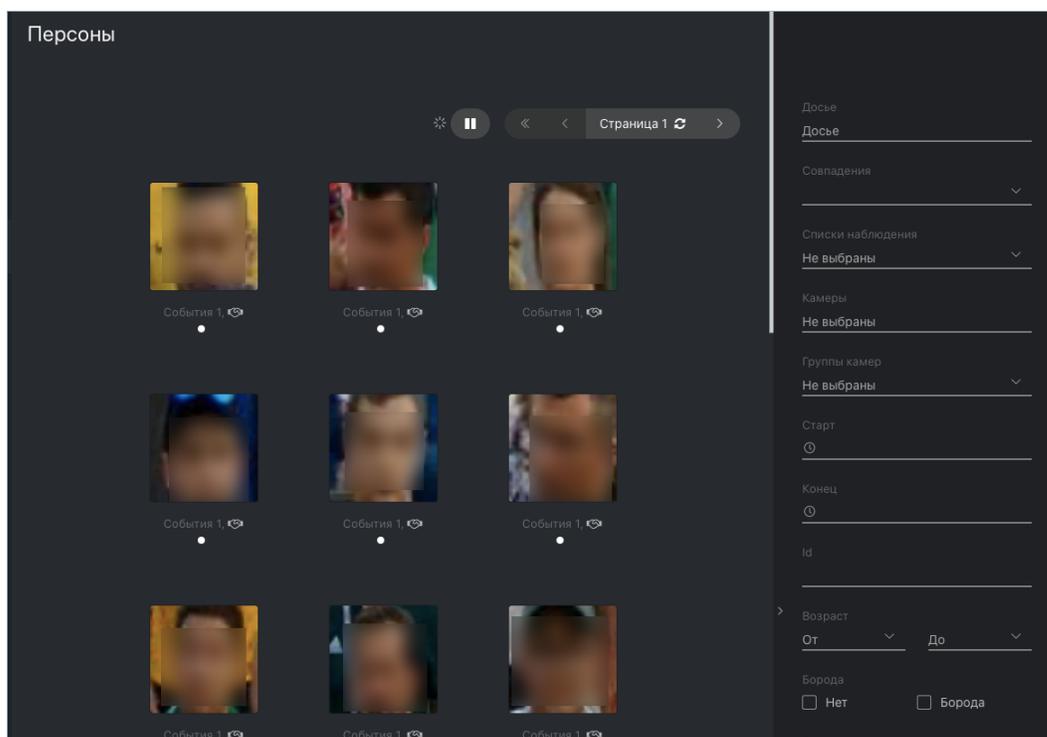
- PERSONS_CLUSTERIZATION_SCHEDULE: рекуррентное правило (RRULE), задающее расписание плановой кластеризации персон.

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 галереей 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*

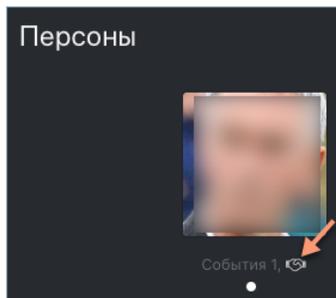
Анализ социальных взаимодействий

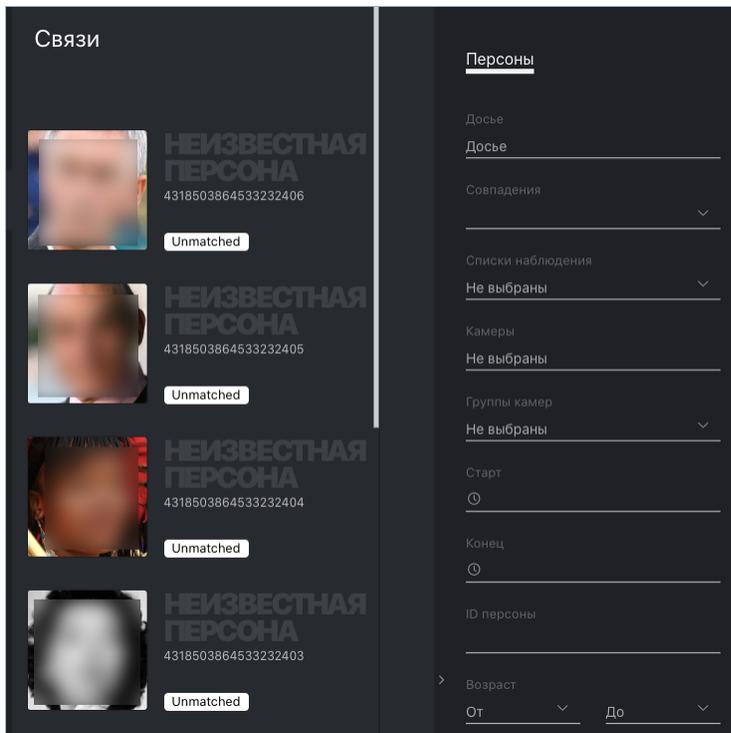
Вы можете посмотреть круг людей, с которыми человек до этого контактировал. Для каждого человека из первого круга система ищет еще один круг связанных с ним людей и т. д. В целом, социальное взаимодействие анализируется до трех кругов в глубину.

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

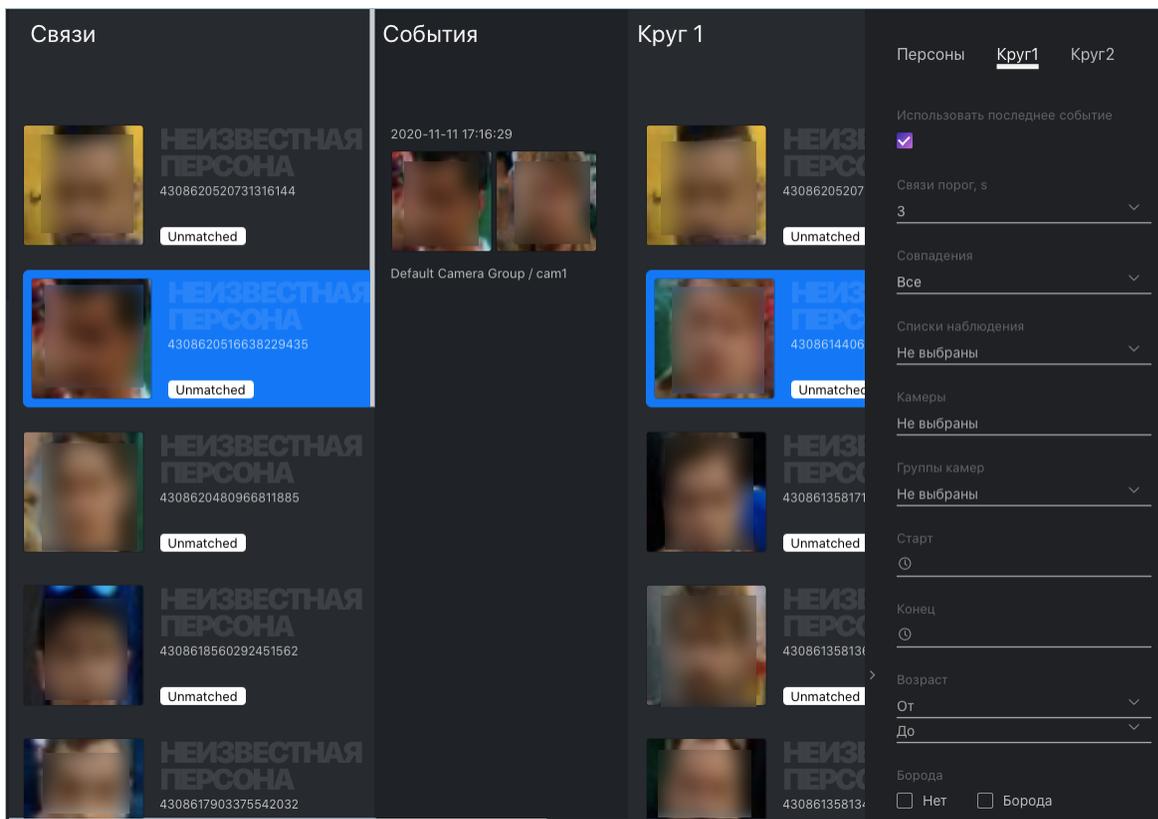
Анализ социальных взаимодействий доступен на вкладке *Связи*.

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.

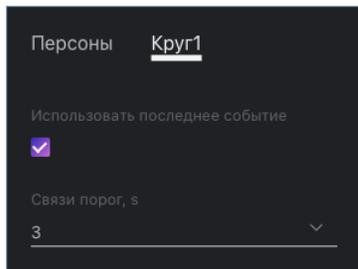


Фильтры можно применять к каждому кругу.

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.

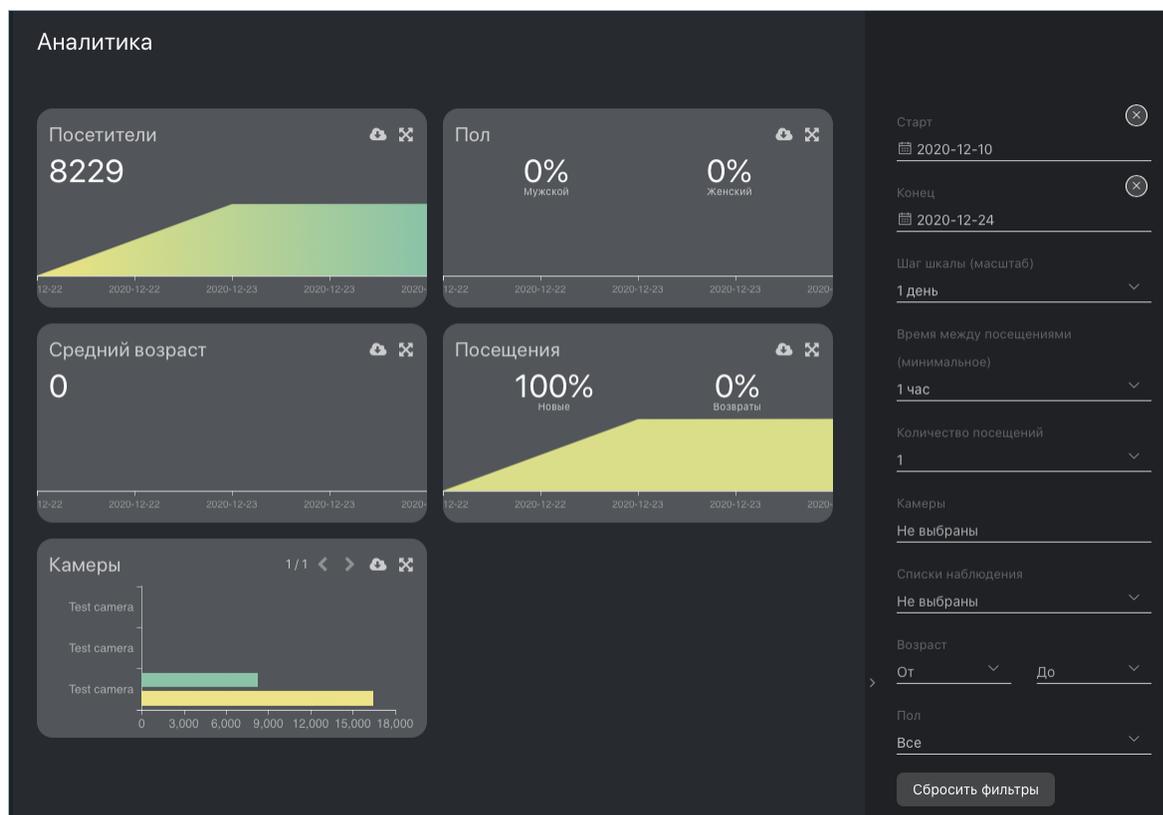


Видеоаналитика

Видеоаналитика ПК FindFace 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

ПК FindFace позволяет строить отчеты по следующим системным объектам:

- события распознавания лиц
- эпизоды
- события поиска
- персоны

- камеры
- досье
- аналитические данные

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.

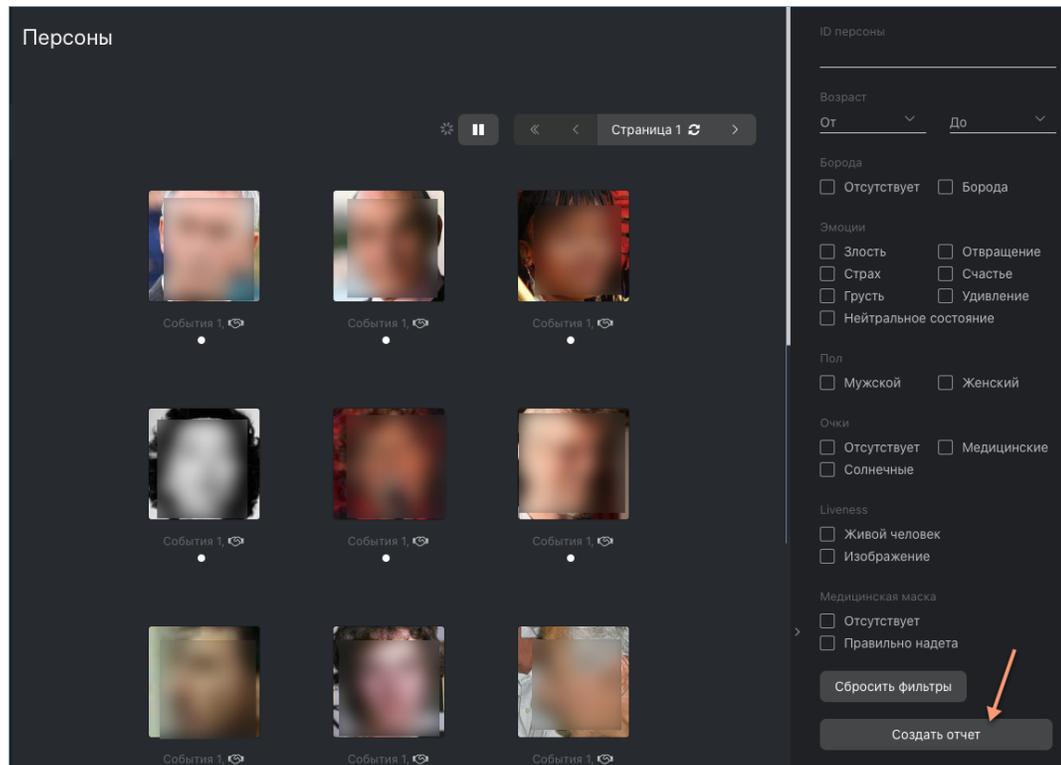
```
sudo vi /etc/findface-security/config.py

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

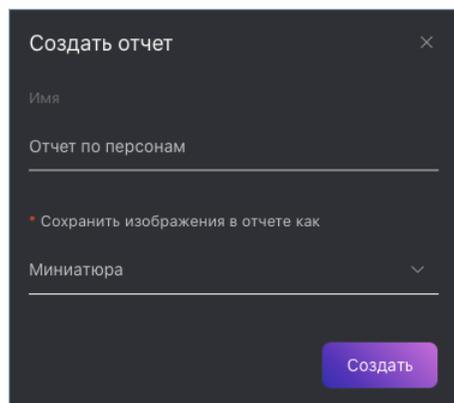
Построение отчета

Для того чтобы построить отчет, выполните следующие действия:

1. Перейдите на вкладку, соответствующую нужному системному объекту: *События, Эпизоды, Поиск, Персоны, Камеры, Досье* или *Аналитика*.
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	10.04КВ	Завершено Скачать
<input type="checkbox"/>	32	Отчет по персонам	Персоны	2020-12-23 19:21:28	2	6.18КВ	Завершено Скачать
<input type="checkbox"/>	30	Отчет по событиям	События	2020-12-22 23:56:29	107	16.82КВ	Завершено Скачать
<input type="checkbox"/>	29	Отчет по событиям	События	2020-12-22 23:55:53	107	16.82КВ	Завершено Скачать
<input type="checkbox"/>	28	Отчет по поиску	События	2020-12-22 23:54:58	10	7.05КВ	Завершено Скачать
<input type="checkbox"/>	26	Отчет по поиску	События	2020-12-22 23:53:02	31	9.33КВ	Завершено Скачать
<input type="checkbox"/>	23	Отчет по событиям	События	2020-12-22 23:50:12	0	5.48КВ	Завершено Скачать
<input type="checkbox"/>	22	Отчет по событиям	События	2020-12-22 23:46:56	68	12.91КВ	Завершено Скачать
<input type="checkbox"/>	21	Отчет по персонам	Персоны	2020-12-22 23:45:56	9	22.25КВ	Завершено Скачать

2.2.12 Видеостена

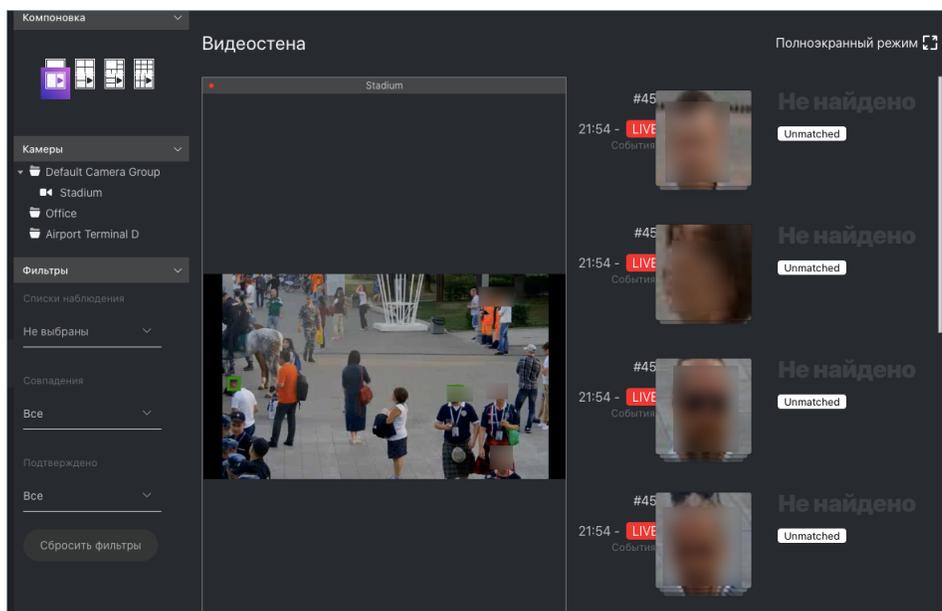
В ПК FindFace встроен базовый функционал видеонаблюдения. Используйте видеостену для отображения видео с камер и видеофайлов.

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.*

2.2.13 Мобильное приложение ПК FindFace

Для оперативной работы с ПК FindFace можно использовать одноименное мобильное приложение. Приложение поставляется для Android по запросу.

В этом разделе:

- *Рекомендуемые системные требования*
- *Аутентификация*
- *Настройки*
- *Функции*
- *Журнал приложения*

Рекомендуемые системные требования

- ОС: Android 6.

Примечание: Функционирование на версиях 4.4-5.x возможно, но не гарантируется.

- CPU: 2 ядра.
- RAM: 2 GB.
- Емкость аккумулятора: не менее 4000 мАч для 8-часовой рабочей смены.

Аутентификация

Для входа в мобильное приложение введите свой логин и пароль ПК FindFace и адрес сервера ПК FindFace.

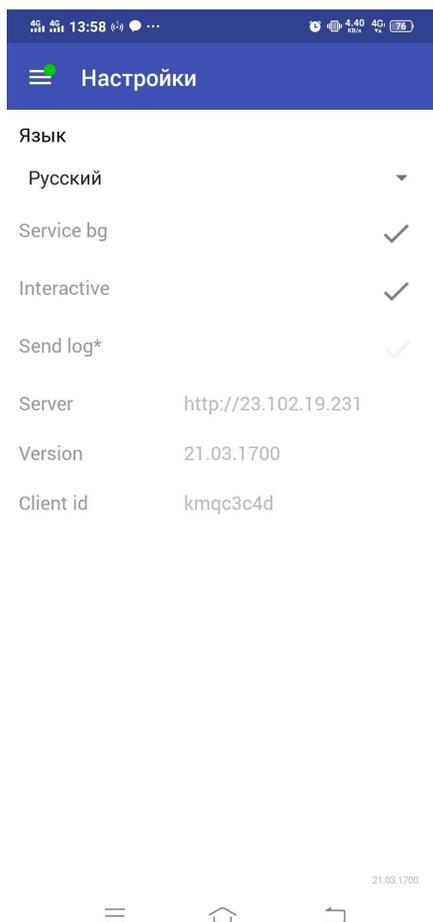
Важно: По умолчанию приложение не доверяет самоподписанным SSL-сертификатам. Если сервер ПК FindFace использует такой сертификат, перед тем как к нему подключиться, поставьте флажок *Self-signed SSL certs* и перезапустите приложение.

The screenshot shows the 'FindFace Security' login interface. At the top, the status bar displays the time 15:59, signal strength, and battery level at 40%. The app title 'FindFace Security' is centered. Below it are three input fields: 'Login' with a blurred value, 'Password' with masked characters, and 'Server address' with the value 'http://23.102.19.231'. A checkbox labeled 'Self-signed SSL certs' is checked, indicated by a checkmark. A blue 'LOGIN' button is positioned at the bottom. At the very bottom of the screen, the Android navigation bar shows the home, back, and recent apps icons, with the time 21.03.1700 displayed.

Настройки

Для настройки мобильного приложения выполните следующие действия:

1. В главном меню нажмите *Настройки*.
2. Для того чтобы открыть настройки, введите PIN-код (по умолчанию 1234).
3. При необходимости переключите язык приложения.



4. Для того чтобы приложение продолжало работать после переключения в фоновый режим, поставьте флажок *Service bg* (по умолчанию активирован).
5. Для того чтобы добавить в приложение режим интерактивных очков, поставьте флажок *Interactive*. В этом режиме приложение будет отправлять информацию о новых совпадениях лиц на умные очки, подключенные к вашему смартфону. Если вам нужен этот функционал, обязательно предварительно свяжитесь со своим менеджером из-за возможных ограничений на использование.
6. Поставьте флажок *Send log*, чтобы отправлять логи приложения на сервер ПК FindFace.

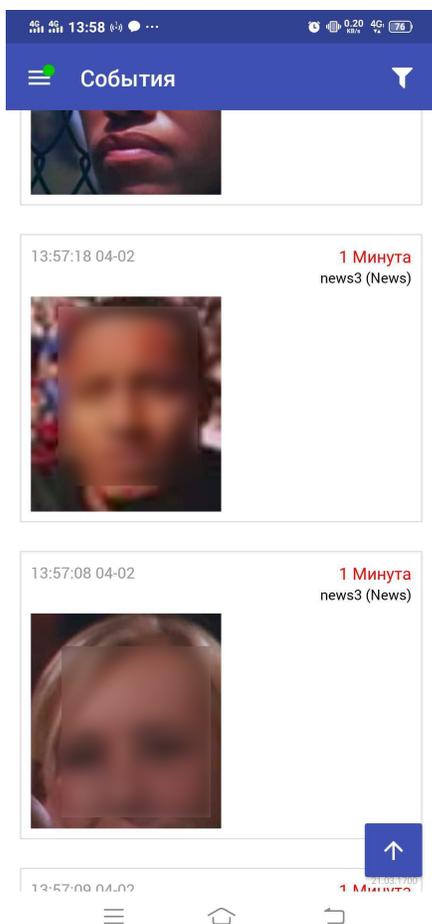
Функции

Мобильный веб-интерфейс имеет удобный и интуитивный дизайн и обеспечивает доступ к следующим оперативным функциям:

- Просмотр и создание досье на персону.

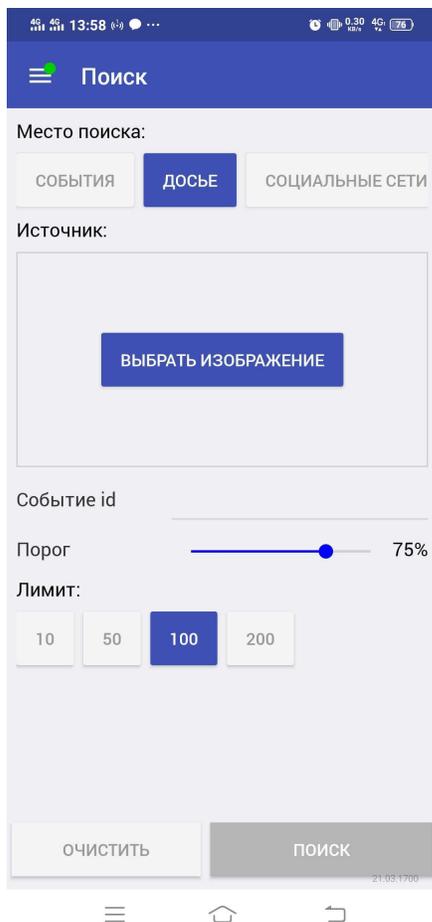


- Получение оповещений о событиях идентификации лиц.



Важно: Для того чтобы получать события в формате push-оповещений, для соответствующего списка наблюдения нужно поставить флажки *Требовать подтверждение события* и *Включить звуковое оповещение* (через полноформатный веб-интерфейс).

- Поиск лиц в базах данных.



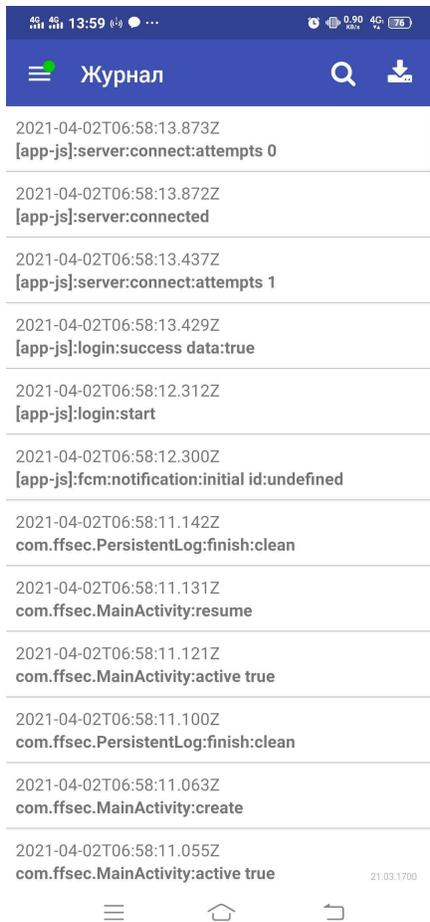
Работа с данными функциями аналогична полноформатной версии.

См.также:

- *Создание досье вручную*
- *Работа с событиями*
- *Поиск по базам данных*

Журнал приложения

В случае сбоя приложения перейдите на вкладку *Журнал*. Экспортируйте журнал приложения и отправьте его в нашу службу техподдержки (support@ntechlab.com).



2.3 Расширенный функционал

2.3.1 Привязка группы камер к экземпляру findface-video-worker

Часто в распределенной архитектуре обработку видеопотока 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 ПК FindFace. 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 в списке наблюдения будущего slave-сервера уже есть досье, они будут заменены на досье с master-сервера во время первой синхронизации. Вся информация в предыдущих досье будет потеряна.

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 ПК FindFace.
- `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
```

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

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

3. Выполните аналогичные действия на каждом slave-сервере.
4. На каждом сервере выполните перенос основной архитектуры базы данных из ПК FindFace 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 биометрической базе данных Tarantool, см. *Пользовательские метаданные лиц досье в Tarantool*.

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

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

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

3. В разделе `FFSECURITY` раскомментируйте секцию `CUSTOM_FIELDS` и измените ее примерное содержание 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. Вы увидите, что новая форма досье появилась в веб-интерфейсе.

2.3.4 Пользовательские метаданные лиц досье в Tarantool

Часто бывает необходимо задать в биометрической 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 -> dossier_face 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': {
...
  'dossier_face': {
    '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 `dossier-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 ПК FindFace утилиту `findface-security-uploader`. Используйте утилиту, когда требуется загрузить большое количество фотографий (более 10000).

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
```

Опции `run`:

- `-parallel`: количество потоков загрузки фотографий, по умолчанию 10. Чем больше потоков, тем быстрее будет завершена загрузка, однако также потребуются 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/`.
- `-user`: имя пользователя.
- `-password`: пароль.
- `-dossier-lists`: перечень разделенных запятой id списков наблюдения, в которые нужно добавить фотографии.
- `-failed`: в случае неудачи при обработке `job`-файла исправьте ошибку и повторите попытку с данной опцией.

2.3.6 Дедупликация событий

В этом разделе:

- *Включение дедупликации*
- *Алгоритм работы дедупликации*

Рассмотрите возможность включения дедупликации, чтобы исключить дублирование событий распознавания лиц 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 лиц. 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:

1. В буферном режиме за один сеанс отслеживания на камере сервер получает одно, наилучшее, изображение лица (сеанс отслеживания продолжается до тех пор, пока лицо не исчезнет из поля зрения камеры).
2. Если в пределах одной группы камер произошло несколько сеансов отслеживания на одной или нескольких камерах в течение указанного интервала дедупликации, ПК FindFace обрабатывает полученные изображения лиц следующим образом:
 - Если в течение предшествующего периода, равного интервалу дедупликации, есть совпадение с досье, ПК FindFace удаляет вновь полученное изображение. В противном случае изображение сохраняется в базе данных.
 - При выполнении дедупликации не совпавших с досье лиц ПК FindFace учитывает как сходство между лицами на различных изображениях, так и качество лиц. ПК FindFace удаляет все изображения похожих лиц в пределах интервала дедупликации, если они более низкого качества, чем первое в данном интервале. Если новое лицо более высокого качества, оно сохраняется. Это гарантирует, что система дедуплицирует события, не пропуская высококачественные лица, необходимые для видеоаналитики.

2.3.7 Распознавание живых лиц в реальном времени (Liveness)

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

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

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

В этом разделе:

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

Включение Liveness-детектора

Для включения Liveness-детектора выполните следующие действия:

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

(continues on next page)

(продолжение с предыдущей страницы)

```
fnc = /usr/share/findface-data/models/faceattr/liveness.alley.v2.cpu.fnc

## 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.fnc
```

2. Перезапустите `findface-video-worker`.

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

Настройка порога Liveness

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

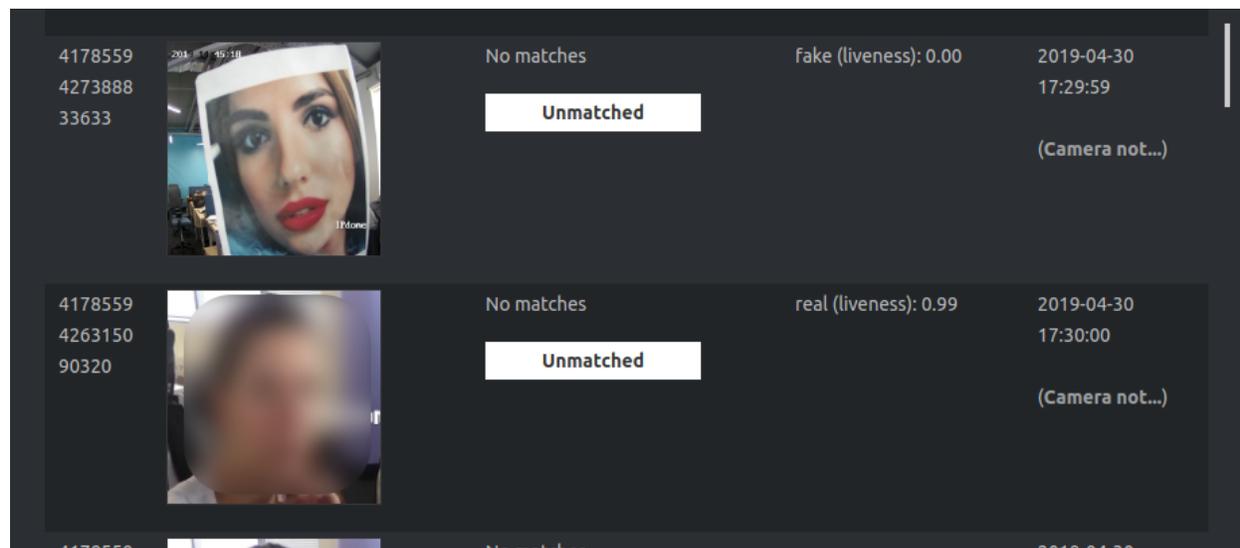
Примечание: Значение по умолчанию является оптимальным. Перед изменением порога проконсультируйтесь у наших специалистов по адресу support@ntechlab.com.

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

'LIVENESS_THRESHOLD': 0.85,
```

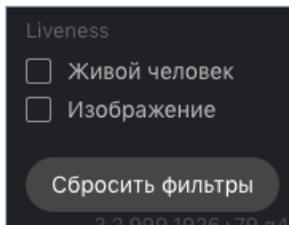
Информация о живом лице в веб-интерфейсе

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



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

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



См.также:

Liveness как автономный сервис

2.3.8 Liveness как автономный сервис

См.также:

Распознавание живых лиц в реальном времени (Liveness)

Помимо *интегрированной* антиспуфинговой системы, отличающей живые лица от их изображений, ПК FindFace 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`.

ПК FindFace 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 ПК FindFace.

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

2.3.9 Распознавание атрибутов лица

ПК FindFace позволяет распознавать в реальном времени такие атрибуты лица, как пол, возраст, эмоции, очки, борода и медицинская маска. Данный функционал доступен как на GPU-, так и на CPU-видеодетекторе лиц.

В этом разделе:

- *Включение распознавания атрибутов лица*
- *Отображение результатов распознавания атрибутов в событиях*
- *Атрибуты лиц в событиях*

Включение распознавания атрибутов лица

Важно: На данном шаге включается распознавание атрибутов лица на уровне HTTP API.

Для того чтобы включить автоматическое распознавание атрибутов лица, откройте файл конфигурации `/etc/findface-extraction-api.ini` и включите соответствующие модели: пол, возраст, эмоции, очки, борода и медицинская маска. 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.

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

models:
  age: faceattr/age.v1.cpu.fnk
  emotions: faceattr/emotions.v1.cpu.fnk
  face: face/jackfruit_480.cpu.fnk
  gender: faceattr/gender.v2.cpu.fnk
  beard: faceattr/beard.v0.cpu.fnk
  glasses3: faceattr/glasses3.v0.cpu.fnk
  medmask3: faceattr/medmask3.v2.cpu.fnk
```

Доступны следующие модели:

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.alleyn.v2.gpu.fnk↵
↵quality.v1.cpu.fnk quality.v1.gpu.fnk
```

Атрибут лица	Ускорение	Параметр в файле конфигурации
биометрия лица	CPU	face: face/ifruit_320.cpu.fnk face: face/jackfruit_160.cpu.fnk face: face/jackfruit_320.cpu.fnk face: face/jackfruit_480.cpu.fnk
	GPU	face: face/ifruit_320.gpu.fnk face: face/jackfruit_160.gpu.fnk face: face/jackfruit_320.gpu.fnk face: face/jackfruit_480.gpu.fnk
возраст	CPU	age: faceattr/age.v1.cpu.fnk
	GPU	age: faceattr/age.v1.gpu.fnk
пол	CPU	gender: faceattr/gender.v2.cpu.fnk
	GPU	gender: faceattr/gender.v2.gpu.fnk
эмоции	CPU	emotions: faceattr/emotions.v1.cpu.fnk
	GPU	emotions: faceattr/emotions.v1.gpu.fnk
очки	CPU	glasses3: faceattr/glasses3.v0.cpu.fnk
	GPU	glasses3: faceattr/glasses3.v0.gpu.fnk
борода	CPU	beard: faceattr/beard.v0.cpu.fnk
	GPU	beard: faceattr/beard.v0.gpu.fnk
медицинская маска	CPU	medmask3: faceattr/medmask3.v2.cpu.fnk
	GPU	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.

```
models:
  gender: ""
  age: ""
  emotions: ""
```

Перезапустите `findface-extraction-api`.

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

После включения моделей обязательно *настройте* отображение атрибутов в веб-интерфейсе.

Отображение результатов распознавания атрибутов в событиях

Для отображения результатов распознавания атрибутов лица в списке событий, добавьте следующую строку в секцию `FFSECURITY: 'EVENTS_FEATURES':` [`'gender'`, `'age'`, `'emotions'`, `'beard'`, `'glasses'`, `'medmask'`], в зависимости от того, какие модели были включены.

Предупреждение: Данная строка должна быть расположена между строками `SF_API_ADDRESS` и `LIVENESS_THRESHOLD`, как показано в примере ниже.

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

(continues on next page)

(продолжение с предыдущей страницы)

```
...
FFSECURITY = {
...
'SF_API_ADDRESS': 'http://127.0.0.1:18411',
'EVENTS_FEATURES': ['gender', 'age', 'emotions', 'beard', 'glasses', 'medmask'],
'LIVENESS_THRESHOLD': 0.85,
'BEARD_THRESHOLD': 0.7,
}
```

Перезапустите `findface-security`.

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

Атрибуты лиц в событиях

После настройки распознавания атрибутов лиц результат распознавания будет выводиться для каждого обнаруженного лица в следующем формате:

Атрибут лица	Формат результата	Пример
Возраст	Атрибут: возраст: количество лет	возраст: 33
Пол	Результат: мужской/женский (атрибут: пол): уверенность алгоритма в результате	женский (пол): 0.95
Эмоции	Результат: злость/отвращение/страх/счастье/грусть/удивление (атрибут: эмоции): уверенность алгоритма в результате	счастье (эмоции): 0.99
Очки	Результат: медицинские/солнечные/нет (атрибут: очки): уверенность алгоритма в результате	нет (очки): 0.87
Борода	Результат: борода/нет (атрибут: борода): уверенность алгоритма в результате	нет (борода): 0.91
Медицинская маска	Результат: нет/есть/неправильно (атрибут: медицинская маска): уверенность алгоритма в результате	нет (медицинская маска): 0.93

При необходимости выполните фильтрацию событий по атрибутам лица.

Возраст
От До

Борода
 Отсутствует Борода

Эмоции
 Злость Отвращение
 Страх Счастье
 Грусть Удивление
 Нейтральное состояние

Пол
 Мужской Женский

Очки
 Отсутствует Медицинские
 Солнечные

Liveness
 Живой человек
 Изображение

Медицинская маска
 Отсутствует
 Неправильно надета
 Правильно надета

2.3.10 Детектирование силуэтов

Для сбора *статистики* человеческих силуэтов в реальном времени необходимо предварительно включить их детектирование.

В этом разделе:

- *Включение детектирования силуэтов*

Включение детектирования силуэтов

Для включения детектирования человеческих силуэтов выполните следующие действия:

1. Откройте файл конфигурации `/etc/findface-extraction-api.ini` и добавьте новый детектор `edie` (`edie_rc2.cpu.fnk/edie_rc2.gpu.fnk`) с соответствующими настройками, как показано в примере ниже.

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

detectors:
  max_batch_size: 1
  instances: 1
  models:
    cheetah:
      model: facedet/cheetah.cpu.fnk
      options:
```

(continues on next page)

(продолжение с предыдущей страницы)

```

min_object_size: 32
resolutions:
- 256x256
- 384x384
- 512x512
- 768x768
- 1024x1024
- 1536x1536
- 2048x2048
edie:
  model: pedet/edie_rc2.cpu.fnk
  options:
    min_object_size: 16
    resolutions:
      - 1280x720
      - 1920x1080
    
```

Примечание: Значения опций детектора `edie`, приведенные в примере, считаются оптимальными. Вы можете настроить их в зависимости от сцен наблюдения используемых камер. Свяжитесь с нашими специалистами по адресу support@ntechlab.com, если у вас есть вопросы.

2. Перезапустите `findface-extraction-api`.

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

2.3.11 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 биометрических

образцов. 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
```

- 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 webservice port, 0=disabled
## type:number env:CFG_STREAMER_PORT longopt:--streamer-port
port = 18999
...

```

- 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

```

- Для применения изменений перезагрузите демон `systemd`.

```
sudo systemctl daemon-reload

```

- 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

```

- Запустите новый сервис.

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

```

- 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

```

(continues on next page)

(продолжение с предыдущей страницы)

```
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
```

2.3.12 Прямые API-запросы к базе данных Tarantool

Вы можете использовать HTTP API для извлечения напрямую данных из базы данных Tarantool.

В этом разделе:

- *Общие сведения*
- *Добавление лица*
- *Удаление лица*
- *Поиск лица*
- *Редактирование метаданных и/или вектора признаков*
- *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 ПК FindFace 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.

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 лица

```
POST /:ver/faces/add/:name
```

Параметры в теле:

Массив лиц в представлении JSON со следующими полями:

- `"id"`: id лица в галерее, `uint64_t`,
- `"facen"`: необработанный вектор признаков, `base64`,
- `"meta"`: метаданные лица, 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=",
```

(continues on next page)

(продолжение с предыдущей страницы)

```
"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 лица

```
POST /v2/faces/delete/:name
```

Параметры в теле:

Массив в представлении JSON из списка id лиц, подлежащих удалению.

Возвращает:

- HTTP 200 и пустое тело в случае успеха.
- HTTP 404, если лицо с заданным id не найдено 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 лица

```
POST /v2/faces/search/:name
```

Параметры в теле:

Поисковый запрос в представлении JSON со следующими полями:

- **limit**: максимальное количество лиц 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 качества лица, поддерживаются только запросы с правой границей 1 (максимальное качество).
 - **id** и **meta/<meta_key>**: фильтры по пользовательскому **id** лиц 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 лицами. Значение в заголовке **X-search-stat** показывает, был ли использован быстрый индекс для поиска: **with_index** или **without_index**.

Примечание: В API v2 быстрый индекс не используется.

- HTTP с отличным от 200 статусом и описание ошибки в теле в случае неудачи.

Пример

Запрос

```
curl -D - -s 'http://localhost:8101/v2/testgal/search' --data '{
  "limit": 2,
  "sort": {
    "score": -1
  },
  "filter": {
    "facen": {
      "data": "qgI3vZRv/zOBQTk9rcirOyZrNp09MdHavW1WuT0=",
      "score": [0.75, 1]
    },
    "id": {
      "range": [922337203685400000, 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

```
POST /v2/faces/update/:name
```

Параметры в теле:

Массив лиц в представлении JSON со следующими полями:

- "id": id лица, 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 лица, в том числе неизменными, в случае успеха.
- HTTP 404 с описанием ошибки, если лица 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 число лиц (`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
```

2.3.13 Включение защиты персональных данных

В ПК FindFace 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
```

Руководство по интеграции

Данная глава посвящена возможностям интеграции с ПК FindFace. Для интеграции своей системы используйте HTTP API, веб-хуки и плагины. Также обратите внимание на список наших готовых интеграций с партнерами.

3.1 HTTP API

Подробная интерактивная документация HTTP API ПК FindFace 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*.

FindFace Security API doc

Internal API documentation

[Base URL: 172.17.46.22 /]
/swagger.json

Authentication
All API methods require a simple token-based HTTP Authentication. In order to authenticate, you should put word "Token" and a key into the Authorization HTTP header, separated by a whitespace:
"Authorization: Token be94403fb59c305b8d6db7ea1f90e019bef3ac85389cf2b10e04b8cf495b31a3"
All requests that fail to provide a valid authentication token will result in a HTTP 401 Unauthorized response.

Parameters Format
There are two ways to pass parameters to the API methods:

- application/json: parameters are represented by a JSON contained in the body.
- multipart/form-data: parameters are encoded into separate parts. This way supports uploading a photo image file in the same request.

Additional Information
Standard extraction limits:

- Image formats: JPEG, PNG, WEBP
- Maximum photo file size: 10 MB
- Maximum photo resolution: 6000 pixels on the biggest side
- Minimal size of a face: 50x50 pixels

Check `/etc/findface-extraction-api.ini` for custom definition
[Contact the developer](#)

Schemes
HTTP

Authorize

auth System Preferences >

3.2 Вебхуки

Вы можете настроить ПК FindFace 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 ПК FindFace 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)*

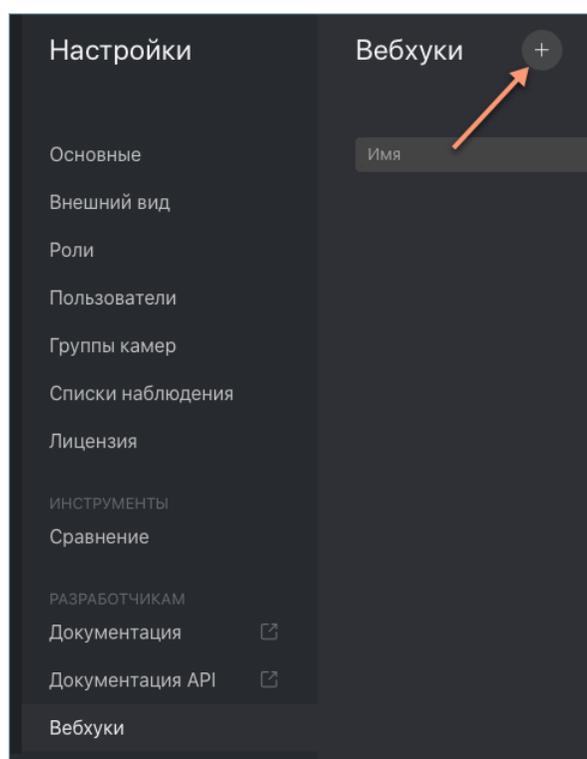
3.2.1 Настройка вебхука

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

```
{
  "events": {
    "allowed_bs_types": [
      "overall",
      "realtime"
    ],
    "camera_group_in": [],
    "camera_in": []
  }
}
```

Активный

Сохранить Назад

4.4

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. ПК FindFace будет автоматически отправлять оповещения о событиях, эпизодах и показаниях

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

События:

- `allowed_bs_types`: режим отслеживания лиц 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.

```
{
  "events": {
    "allowed_bs_types": [
      "overall",
      "realtime"
    ],
    "camera_group_in": [],
```

(continues on next page)

(продолжение с предыдущей страницы)

```

        "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": 0,
        "faces_lte": 0,
        "silhouettes_gte": 0,
        "silhouettes_lte": 0
    }
}

```

Важно: Используйте только фильтры, соответствующие цели поиска. Для исключения фильтра удалите его из вебхука. Не оставляйте фильтр пустым ({}), поскольку в этом случае фильтр вернет пустой результат.

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

```
{}
```

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

```
{ "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)])

# 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 ПК FindFace, 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

```
[{'acknowledged': True,
  'acknowledged_by': None,
  'acknowledged_date': '2020-05-18T15:08:38+00:00',
  'acknowledged_reaction': '',
  'bs_type': 'overall',
  'camera': None,
  'camera_group': 1,
  'confidence': 0.0,
  'created_date': '2020-05-18T15:08:38+00:00',
  'episode': None,
  'event_type': 'event_created',
  'face': 'http://172.17.46.134/uploads/2020/05/18/event/150842_face_AgohWm.jpg',
  'features': {'age': None,
               'beard': None,
               'emotions': None,
               'gender': None,
               'glasses': None,
               'liveness': None,
               'medmask': None},
  'frame': 'http://172.17.46.134/uploads/2020/05/18/event/150842_full_frame_Y3vtGe.jpg',
  'frame_coords_bottom': 320,
  'frame_coords_left': 117,
  'frame_coords_right': 170,
  'frame_coords_top': 242,
  'id': '4267625862518432158',
  'looks_like_confidence': None,
  'matched': False,
  'matched_dossier': None,
  'matched_face': '',
  'matched_lists': [-1],
  'quality': -0.000766,
  'scores': {'liveness_score': None,
              'quality': -0.000766480341553,
              'track': {'first_timestamp': '2020-05-18T15:08:38',
                        'id': '43277e17b1c2-44',
                        'last_timestamp': '2020-05-18T15:08:39'},
              'track_duration_seconds': 2.502499999999997},
  'video_source': 1,
  'webhook_type': 'events'}}
```

Открытие эпизода

```
[{'acknowledged': True,
  'acknowledged_by': None,
  'acknowledged_date': None,
  'acknowledged_reaction': '',
  'best_event': '4267637154774219594',
  'camera_groups': [1],
  'cameras': []},
```

(continues on next page)

(продолжение с предыдущей страницы)

```
'closed_date': None,
'created_date': '2020-05-18T16:18:49.111880Z',
'event_type': 'episode_open',
'events_count': 1,
'features': None,
'id': 2118,
'last_event': {'acknowledged': True,
               'acknowledged_by': None,
               'acknowledged_date': '2020-05-18T16:18:46+00:00',
               'acknowledged_reaction': '',
               'camera': None,
               'camera_group': 1,
               'confidence': 0.0,
               'created_date': '2020-05-18T16:18:46+00:00',
               'episode': 2118,
               'face': 'http://172.17.46.134/uploads/2020/05/18/event/161849_face_
↔j2TQwk.jpg',
               'features': {'age': None,
                           'beard': None,
                           'emotions': None,
                           'gender': None,
                           'glasses': None,
                           'liveness': None,
                           'medmask': None},
               'frame': 'http://172.17.46.134/uploads/2020/05/18/event/161849_full_
↔frame_vTfuH9.jpg',
               'frame_coords_bottom': 327,
               'frame_coords_left': 778,
               'frame_coords_right': 901,
               'frame_coords_top': 161,
               'id': '4267637154774219594',
               'looks_like_confidence': None,
               'matched': False,
               'matched_dossier': None,
               'matched_face': '',
               'matched_lists': [-1],
               'quality': -0.000311,
               'scores': {'liveness_score': None,
                          'quality': -0.000311948591843,
                          'track': {'first_timestamp': '2020-05-18T16:18:46',
                                    'id': '1ee9a3612af3-9',
                                    'last_timestamp': '2020-05-18T16:18:47'},
                          'track_duration_seconds': 2.0399999999999999},
               'video_source': 2},
'matched_event': None,
'matched_lists': [-1],
'open': True,
'webhook_type': 'episodes']}]
```

Заккрытие эпизода

```
[{'acknowledged': True,
  'acknowledged_by': None,
  'acknowledged_date': None,
  'acknowledged_reaction': '',
  'best_event': {'acknowledged': True,
                 'acknowledged_by': None,
                 'acknowledged_date': '2020-05-18T15:09:57+00:00',
                 'acknowledged_reaction': '',
                 'camera': None,
                 'camera_group': 1,
                 'confidence': 0.0,
                 'created_date': '2020-05-18T15:09:57+00:00',
                 'episode': 518,
                 'face': 'http://172.17.46.134/uploads/2020/05/18/event/151012_face_
↪5LLHQL.jpg',
                 'features': {'age': None,
                              'beard': None,
                              'emotions': None,
                              'gender': None,
                              'glasses': None,
                              'liveness': None,
                              'medmask': None},
                 'frame': 'http://172.17.46.134/uploads/2020/05/18/event/151012_full_
↪frame_CdNn2N.jpg',
                 'frame_coords_bottom': 299,
                 'frame_coords_left': 917,
                 'frame_coords_right': 1005,
                 'frame_coords_top': 179,
                 'id': '4267626103667833809',
                 'looks_like_confidence': None,
                 'matched': False,
                 'matched_dossier': None,
                 'matched_face': '',
                 'matched_lists': [-1],
                 'quality': -0.653877,
                 'scores': {'liveness_score': None,
                            'quality': -0.653877139091491,
                            'track': {'first_timestamp': '2020-05-18T15:09:57',
                                       'id': '43277e17b1c2-231',
                                       'last_timestamp': '2020-05-18T15:09:57'},
                            'track_duration_seconds': 0.250255555555554},
                 'video_source': 1},
  'camera_groups': [1],
  'cameras': [],
  'closed_date': '2020-05-18T15:10:42.870851Z',
  'created_date': '2020-05-18T15:10:12.201230Z',
  'event_type': 'episode_close',
  'events_count': 1,
  'features': None,
  'id': 518,
  'last_event': {'acknowledged': True,
```

(continues on next page)

(продолжение с предыдущей страницы)

```

'acknowledged_by': None,
'acknowledged_date': '2020-05-18T15:09:57+00:00',
'acknowledged_reaction': '',
'camera': None,
'camera_group': 1,
'confidence': 0.0,
'created_date': '2020-05-18T15:09:57+00:00',
'episode': 518,
'face': 'http://172.17.46.134/uploads/2020/05/18/event/151012_face_
↔5L1HQL.jpg',
'features': {'age': None,
             'beard': None,
             'emotions': None,
             'gender': None,
             'glasses': None,
             'liveness': None,
             'medmask': None},
'frame': 'http://172.17.46.134/uploads/2020/05/18/event/151012_full_
↔frame_CdNn2N.jpg',
'frame_coords_bottom': 299,
'frame_coords_left': 917,
'frame_coords_right': 1005,
'frame_coords_top': 179,
'id': '4267626103667833809',
'looks_like_confidence': None,
'matched': False,
'matched_dossier': None,
'matched_face': '',
'matched_lists': [-1],
'quality': -0.653877,
'scores': {'liveness_score': None,
           'quality': -0.653877139091491,
           'track': {'first_timestamp': '2020-05-18T15:09:57',
                    'id': '43277e17b1c2-231',
                    'last_timestamp': '2020-05-18T15:09:57'},
           'track_duration_seconds': 0.250255555555554},
'video_source': 1},
'matched_event': None,
'matched_lists': [-1],
'open': False,
'webhook_type': 'episodes']]

```

Запись счетчика

```
[{'camera': 3,
  'camera_group': 1,
  'counter': 2,
  'counter_name': 'smosh',
  'created_date': '2020-05-18T16:15:06.679592Z',
  'event_type': 'counter_record',
  'faces_bbox': [[[700, 210], [894, 210], [894, 464], [700, 464]],
                 [[160, 190], [304, 190], [304, 394], [160, 394]]],
  'faces_count': 2,
  'fullframe': 'http://172.17.46.134/uploads/2020/05/18/counters/161506_fullframe_7Z8n7X.
↪jpg',
  'id': 16,
  'silhouettes_bbox': [[[15, 135], [584, 135], [584, 709], [15, 709]],
                      [[585, 80], [1194, 80], [1194, 684], [585, 684]],
                      [[0, 380], [69, 380], [69, 714], [0, 714]]],
  'silhouettes_count': 3,
  'thumbnail': 'http://172.17.46.134/uploads/2020/05/18/counters/161506_thumb_XLMFwE.jpg
↪',
  'webhook_type': 'counters'}}
```

Для просмотра статуса отправки вебхука в ПК FindFace, выполните следующую команду:

```
sudo journalctl -u findface-security.service | grep 'WebhooksManager'
```

Ответ в случае успеха:

```
May 18 21:21:38 qa-2 ffsecurity[17851]: INFO      [WebhooksManager] Updating "events"
↪workers for webhooks: {2}
May 18 21:21:52 qa-2 ffsecurity[17851]: INFO      [WebhooksManager] [SC:9KHqkQg7-
↪VW:aa3af58f] Webhook updater processing message(type-"events:event_created"). Consumer
↪reception delta: 0.002617
May 18 21:21:52 qa-2 ffsecurity[17851]: INFO      [WebhooksManager:2] <queue: 0> Webhook
↪worker(id-2, type-"events") sent batch(len-1, type-"events"): ['4267685965791894192']
May 18 21:21:53 qa-2 ffsecurity[17851]: INFO      [WebhooksManager] [SC:v59UsC1V-
↪VW:75c4a9ec] Webhook updater processing message(type-"events:event_created"). Consumer
↪reception delta: 0.002386
May 18 21:21:53 qa-2 ffsecurity[17851]: INFO      [WebhooksManager:2] <queue: 0> Webhook
↪worker(id-2, type-"events") sent batch(len-1, type-"events"): ['4267685968207813297']
May 18 21:21:53 qa-2 ffsecurity[17851]: INFO      [WebhooksManager] [SC:vKNlXiIn-
↪VW:c0219d31] Webhook updater processing message(type-"events:event_created"). Consumer
↪reception delta: 0.004499
May 18 21:21:53 qa-2 ffsecurity[17851]: INFO      [WebhooksManager:2] <queue: 0> Webhook
↪worker(id-2, type-"events") sent batch(len-1, type-"events"): ['4267685968837561053']
May 18 21:21:55 qa-2 ffsecurity[17851]: INFO      [WebhooksManager] [SC:zZ08v4LJ-
↪VW:feff75dd] Webhook updater processing message(type-"events:event_created"). Consumer
↪reception delta: 0.001905
May 18 21:21:55 qa-2 ffsecurity[17851]: INFO      [WebhooksManager:2] <queue: 0> Webhook
↪worker(id-2, type-"events") sent batch(len-1, type-"events"): ['4267685973269790230']
May 18 21:21:57 qa-2 ffsecurity[17851]: INFO      [WebhooksManager] [SC:EbpDe124-
↪VW:083688e2] Webhook updater processing message(type-"events:event_created"). Consumer
```

(continues on next page)

(продолжение с предыдущей страницы)

```

↪reception delta: 0.002017
May 18 21:21:57 qa-2 ffsecurity[17851]: INFO      [WebhooksManager:2] <queue: 0> Webhook_
↪worker(id-2, type-"events") sent batch(len-1, type-"events"): ['4267685977917324748']
May 18 21:21:57 qa-2 ffsecurity[17851]: INFO      [WebhooksManager] [SC:L5XoQTdq-
↪VW:6f1e397f] Webhook updater processing message(type-"events:event_created"). Consumer_
↪reception delta: 0.009237
May 18 21:21:57 qa-2 ffsecurity[17851]: INFO      [WebhooksManager:2] <queue: 0> Webhook_
↪worker(id-2, type-"events") sent batch(len-1, type-"events"): ['4267685979796372941']
May 18 21:21:58 qa-2 ffsecurity[17851]: INFO      [WebhooksManager] [SC:ZZ33mwuv-
↪VW:a4cad3a2] Webhook updater processing message(type-"events:event_created"). Consumer_
↪reception delta: 0.008542
May 18 21:21:58 qa-2 ffsecurity[17851]: INFO      [WebhooksManager:2] <queue: 0> Webhook_
↪worker(id-2, type-"events") sent batch(len-1, type-"events"): ['4267685980899116054']
May 18 21:21:58 qa-2 ffsecurity[17851]: INFO      [WebhooksManager] [SC:BfAQRgp0-
↪VW:4c19b207] Webhook updater processing message(type-"events:event_created"). Consumer_
↪reception delta: 0.003183
May 18 21:21:58 qa-2 ffsecurity[17851]: INFO      [WebhooksManager:2] <queue: 0> Webhook_
↪worker(id-2, type-"events") sent batch(len-1, type-"events"): ['4267685982215838395']

```

Ответ, если попытка была неудачной:

```

May 18 21:29:09 qa-2 ffsecurity[17851]: INFO      [WebhooksManager] [SC:jrGdiC7e-
↪VW:2def51cf] Webhook updater processing message(type-"events:event_created"). Consumer_
↪reception delta: 0.003909
May 18 21:29:09 qa-2 ffsecurity[17851]: WARNING  [WebhooksManager:2] <queue: 1> Webhook_
↪worker(id-2, type-"events") Error sending webhook: 405, message='Not Allowed'. Attempt_
↪2 out of 10. Next attempt in 0.729 seconds.
May 18 21:29:10 qa-2 ffsecurity[17851]: INFO      [WebhooksManager] [SC:jgqLszI7-
↪VW:6a7fea19] Webhook updater processing message(type-"events:event_created"). Consumer_
↪reception delta: 0.002402
May 18 21:29:10 qa-2 ffsecurity[17851]: WARNING  [WebhooksManager:2] <queue: 2> Webhook_
↪worker(id-2, type-"events") Error sending webhook: 405, message='Not Allowed'. Attempt_
↪3 out of 10. Next attempt in 1.968 seconds.
May 18 21:29:10 qa-2 ffsecurity[17851]: INFO      [WebhooksManager] [SC:LLGB1RRR-
↪VW:053d7c7d] Webhook updater processing message(type-"events:event_created"). Consumer_
↪reception delta: 0.003794
May 18 21:29:11 qa-2 ffsecurity[17851]: INFO      [WebhooksManager] [SC:4V123NQD-
↪VW:a4640479] Webhook updater processing message(type-"events:event_created"). Consumer_
↪reception delta: 0.037162
May 18 21:29:11 qa-2 ffsecurity[17851]: INFO      [WebhooksManager] [SC:QKY577ed-
↪VW:41cd531a] Webhook updater processing message(type-"events:event_created"). Consumer_
↪reception delta: 0.005274
May 18 21:29:12 qa-2 ffsecurity[17851]: INFO      [WebhooksManager] [SC:SVSrlj1n-
↪VW:973ae0dd] Webhook updater processing message(type-"events:event_created"). Consumer_
↪reception delta: 0.004273
May 18 21:29:12 qa-2 ffsecurity[17851]: WARNING  [WebhooksManager:2] <queue: 6> Webhook_
↪worker(id-2, type-"events") Error sending webhook: 405, message='Not Allowed'. Attempt_
↪4 out of 10. Next attempt in 5.314 seconds.
May 18 21:29:12 qa-2 ffsecurity[17851]: INFO      [WebhooksManager] [SC:pho03HfD-
↪VW:9c6812d1] Webhook updater processing message(type-"events:event_created"). Consumer_

```

(continues on next page)

(продолжение с предыдущей страницы)

```
↪reception delta: 0.019604
May 18 21:29:13 qa-2 ffsecurity[17851]: INFO      [WebhooksManager] [SC:WDMmZ5MO-
↪VW:842b3397] Webhook updater processing message(type-"events:event_created"). Consumer_
↪reception delta: 0.231164
```

3.2.3 Подробные данные в вебхуках (Verbose)

По умолчанию оповещения по вебхуку содержат только идентификаторы таких объектов, как досье, списки наблюдения, камеры и группы камер. Для того чтобы получать подробную информацию по данным объектам, 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 (Verbose)

```
[{'acknowledged': True,
  'acknowledged_by': None,
  'acknowledged_date': '2020-07-30T14:41:52+00:00',
  'acknowledged_reaction': '',
  'bs_type': 'overall',
  'camera': {'active': True,
             'azimuth': None,
             'comment': '',
             'created_date': '2020-07-12T05:57:50.459974Z',
             'group': 1,
             'health_status': {'code': 'yellow',
                               'code_desc': 'Some faces from this camera '
                                           'couldn't be processed',
                               'enabled': True,
                               'msg': '',
                               'statistic': {'decoding_soft_errors': 0,
                                             'faces_failed': 17,
                                             'faces_not_posted': 0,
                                             'faces_posted': 185,
                                             'frames_dropped': 1066,
                                             'frame_height': 1080,
```

(continues on next page)

(продолжение с предыдущей страницы)

```

        'frames_imotion_skipped': 0,
        'frames_processed': 36421,
        'frame_width': 1920
        'job_starts': 1,
        'processed_duration': 1499.44,
        'processing_fps': 38.595585},
    'status': 'INPROGRESS'},
'id': 1,
'latitude': None,
'longitude': None,
'modified_date': '2020-07-20T06:38:58.607160Z',
'name': '1',
'screenshot': 'http://172.17.47.245/cameras/1/screenshot/',
'stream_settings': {'api_timeout': 15000,
                    'draw_track': False,
                    'fd_frame_height': -1,
                    'ffmpeg_format': '',
                    'ffmpeg_params': [],
                    'jpeg_quality': 95,
                    'max_face_size': 0,
                    'md_scale': 0.3,
                    'md_threshold': 0.002,
                    'min_d_score': -1000,
                    'min_face_size': 0,
                    'min_score': -2,
                    'npersons': 4,
                    'overall': True,
                    'realtime': False,
                    'realtime_dly': 500,
                    'realtime_post_perm': False,
                    'roi': '',
                    'rot': '',
                    'tracker_threads': 4},
'stream_settings_gpu': {'ffmpeg_format': '',
                        'ffmpeg_params': [],
                        'filter_max_face_size': 8192,
                        'filter_min_face_size': 1,
                        'filter_min_quality': 0.45,
                        'imotion_threshold': 0,
                        'jpeg_quality': 95,
                        'overall_only': True,
                        'play_speed': -1,
                        'realtime_post_every_interval': False,
                        'realtime_post_first_immediately': False,
                        'realtime_post_interval': 1,
                        'roi': '',
                        'rot': '',
                        'router_timeout_ms': 15000,
                        'router_verify_ssl': True,
                        'start_stream_timestamp': 0,
                        'use_stream_timestamp': False},
'threshold': None,

```

(continues on next page)

(продолжение с предыдущей страницы)

```

        'url': 'http://a3569458063-s26881.cdn.ngenix.net/live/smil:r24.smil/
↳ chunklist_b1200000.m3u8?codec=mpeg4'},
    'camera_group': {'active': True,
                    'comment': '',
                    'created_date': '2020-07-12T05:48:09.537724Z',
                    'deduplicate': True,
                    'deduplicateDelay': 10,
                    'id': 1,
                    'labels': {},
                    'modified_date': '2020-07-17T01:41:22.944825Z',
                    'name': 'Default Camera Group',
                    'permissions': {'1': 'view', '2': 'view', '3': 'view'},
                    'threshold': None},
    'confidence': 0.0,
    'created_date': '2020-07-30T14:41:52+00:00',
    'episode': None,
    'event_type': 'event_created',
    'face': 'http://172.17.47.245/uploads/2020/07/30/event/144203_face_5ks7RN.jpg',
    'features': {'age': None,
                'beard': None,
                'emotions': None,
                'gender': None,
                'glasses': None,
                'liveness': None,
                'medmask': None},
    'frame': 'http://172.17.47.245/uploads/2020/07/30/event/144203_full_frame_OC4sG3.jpg',
    'frame_coords_bottom': 427,
    'frame_coords_left': 367,
    'frame_coords_right': 600,
    'frame_coords_top': 119,
    'id': '4284552331019521692',
    'looks_like_confidence': None,
    'matched': False,
    'matched_dossier': None,
    'matched_face': '',
    'matched_lists': [{'acknowledge': False,
                      'active': True,
                      'camera_groups': [],
                      'color': 'ffffff',
                      'comment': 'Default list for unmatched evenets',
                      'created_date': '2020-07-12T05:48:09.324264Z',
                      'id': -1,
                      'modified_date': '2020-07-12T05:48:09.324369Z',
                      'name': 'Unmatched',
                      'notify': False,
                      'permissions': {},
                      'remote_url': None,
                      'threshold': None}],
    'quality': -0.000112,
    'scores': {'liveness_score': None,
               'quality': -0.00011235895362900001,
               'track': {'first_timestamp': '2020-07-30T14:41:51',

```

(continues on next page)

(продолжение с предыдущей страницы)

```
'id': '313e117d86b3-203',
  'last_timestamp': '2020-07-30T14:42:01'},
  'track_duration_seconds': 4.7999999999999272},
'temperature': None,
'webhook_type': 'events']}]
```

Открытие эпизода (Verbose)

```
[{'acknowledged': True,
  'acknowledged_by': None,
  'acknowledged_date': None,
  'acknowledged_reaction': '',
  'best_event': '4284565234541639834',
  'camera_groups': [{'active': True,
    'comment': '',
    'created_date': '2020-07-12T05:48:09.537724Z',
    'deduplicate': True,
    'deduplicateDelay': 10,
    'id': 1,
    'labels': {},
    'modified_date': '2020-07-17T01:41:22.944825Z',
    'name': 'Default Camera Group',
    'permissions': {'1': 'view', '2': 'view', '3': 'view'},
    'threshold': None}],
  'cameras': [{'active': True,
    'azimuth': None,
    'comment': '',
    'created_date': '2020-07-12T05:57:50.459974Z',
    'group': 1,
    'health_status': {'code': 'yellow',
      'code_desc': 'Some faces from this camera '
        'couldn't be processed',
      'enabled': True,
      'msg': '',
      'statistic': {'decoding_soft_errors': 0,
        'faces_failed': 20,
        'faces_not_posted': 0,
        'faces_posted': 1027,
        'frames_dropped': 4082,
        'frame_height': 1080,
        'frames_imotion_skipped': 0,
        'frames_processed': 153841,
        'frame_width': 1920,
        'job_starts': 1,
        'processed_duration': 6320.04,
        'processing_fps': 49.990253},
      'status': 'INPROGRESS'},
    'id': 1,
    'latitude': None,
    'longitude': None,
```

(continues on next page)

(продолжение с предыдущей страницы)

```
'modified_date': '2020-07-20T06:38:58.607160Z',
'name': '1',
'screenshot': 'http://172.17.47.245/cameras/1/screenshot/',
'stream_settings': {'api_timeout': 15000,
                    'draw_track': False,
                    'fd_frame_height': -1,
                    'ffmpeg_format': '',
                    'ffmpeg_params': [],
                    'jpeg_quality': 95,
                    'max_face_size': 0,
                    'md_scale': 0.3,
                    'md_threshold': 0.002,
                    'min_d_score': -1000,
                    'min_face_size': 0,
                    'min_score': -2,
                    'npersons': 4,
                    'overall': True,
                    'realtime': False,
                    'realtime_dly': 500,
                    'realtime_post_perm': False,
                    'roi': '',
                    'rot': '',
                    'tracker_threads': 4},
'stream_settings_gpu': {'ffmpeg_format': '',
                        'ffmpeg_params': [],
                        'filter_max_face_size': 8192,
                        'filter_min_face_size': 1,
                        'filter_min_quality': 0.45,
                        'imotion_threshold': 0,
                        'jpeg_quality': 95,
                        'overall_only': True,
                        'play_speed': -1,
                        'realtime_post_every_interval': False,
                        'realtime_post_first_immediately': False,
                        'realtime_post_interval': 1,
                        'roi': '',
                        'rot': '',
                        'router_timeout_ms': 15000,
                        'router_verify_ssl': True,
                        'start_stream_timestamp': 0,
                        'use_stream_timestamp': False},
'threshold': None,
'url': 'http://a3569458063-s26881.cdn.ngenix.net/live/smil:r24.smil/
↳ chunklist_b1200000.m3u8?codec=mpeg4'}],
'closed_date': None,
'created_date': '2020-07-30T16:01:52Z',
'event_type': 'episode_open',
'events_count': 1,
'features': None,
'id': 104229,
'last_event': {'acknowledged': True,
               'acknowledged_by': None,
```

(continues on next page)

(продолжение с предыдущей страницы)

```

'acknowledged_date': '2020-07-30T16:01:52+00:00',
'acknowledged_reaction': '',
'camera': 1,
'camera_group': 1,
'confidence': 0.0,
'created_date': '2020-07-30T16:01:52+00:00',
'episode': 104229,
'face': 'http://172.17.47.245/uploads/2020/07/30/event/160210_face_
↔LnGXeX.jpg',
'features': {'age': None,
             'beard': None,
             'emotions': None,
             'gender': None,
             'glasses': None,
             'liveness': None,
             'medmask': None},
'frame': 'http://172.17.47.245/uploads/2020/07/30/event/160210_full_
↔frame_2N1x5Y.jpg',
'frame_coords_bottom': 254,
'frame_coords_left': 223,
'frame_coords_right': 319,
'frame_coords_top': 137,
'id': '4284565234541639834',
'looks_like_confidence': None,
'matched': False,
'matched_dossier': None,
'matched_face': '',
'matched_lists': [-1],
'quality': 0.000206,
'scores': {'liveness_score': None,
           'quality': 0.00020667129138,
           'track': {'first_timestamp': '2020-07-30T16:01:48',
                     'id': '313e117d86b3-1047',
                     'last_timestamp': '2020-07-30T16:02:09'},
           'track_duration_seconds': 19.560000000000131},
'temperature': None,
'video_source': None},
,
'matched_event': None,
'matched_lists': [{'acknowledge': False,
                   'active': True,
                   'camera_groups': [],
                   'color': 'ffffff',
                   'comment': 'Default list for unmatched evenets',
                   'created_date': '2020-07-12T05:48:09.324264Z',
                   'id': -1,
                   'modified_date': '2020-07-12T05:48:09.324369Z',
                   'name': 'Unmatched',
                   'notify': False,
                   'permissions': {},
                   'remote_url': None,
                   'threshold': None}],

```

(continues on next page)

(продолжение с предыдущей страницы)

```
'open': True,
'temperature': None,
'webhook_type': 'episodes']}]
```

Заккрытие эпизода (Verbose)

```
[{'acknowledged': True,
  'acknowledged_by': None,
  'acknowledged_date': None,
  'acknowledged_reaction': '',
  'best_event': {'acknowledged': True,
                 'acknowledged_by': None,
                 'acknowledged_date': '2020-07-30T16:05:09+00:00',
                 'acknowledged_reaction': '',
                 'camera': 1,
                 'camera_group': 1,
                 'confidence': 0.0,
                 'created_date': '2020-07-30T16:05:06+00:00',
                 'episode': 104236,
                 'face': 'http://172.17.47.245/uploads/2020/07/30/event/160509_face_
↪PXGmaZ.jpg',
                 'features': {'age': None,
                              'beard': None,
                              'emotions': None,
                              'gender': None,
                              'glasses': None,
                              'liveness': None,
                              'medmask': None},
                 'frame': 'http://172.17.47.245/uploads/2020/07/30/event/160509_full_
↪frame_YeIRKk.jpg',
                 'frame_coords_bottom': 322,
                 'frame_coords_left': 465,
                 'frame_coords_right': 574,
                 'frame_coords_top': 190,
                 'id': '4284565716150084776',
                 'looks_like_confidence': None,
                 'matched': False,
                 'matched_dossier': None,
                 'matched_face': '',
                 'matched_lists': [-1],
                 'quality': -0.000468,
                 'scores': {'liveness_score': None,
                            'quality': -0.000468814512714,
                            'track': {'first_timestamp': '2020-07-30T16:04:56',
                                       'id': '313e117d86b3-1071',
                                       'last_timestamp': '2020-07-30T16:05:08'},
                            'track_duration_seconds': 8.159999999999854},
                 'temperature': None},
  'camera_groups': [{'active': True,
                    'comment': ''}]
```

(continues on next page)

(продолжение с предыдущей страницы)

```

'created_date': '2020-07-12T05:48:09.537724Z',
'deduplicate': True,
'deduplicateDelay': 10,
'id': 1,
'labels': {},
'modified_date': '2020-07-17T01:41:22.944825Z',
'name': 'Default Camera Group',
'permissions': {'1': 'view', '2': 'view', '3': 'view'},
'threshold': None}],
'cameras': [{ 'active': True,
'azimuth': None,
'comment': '',
'created_date': '2020-07-12T05:57:50.459974Z',
'group': 1,
'health_status': { 'code': 'yellow',
'code_desc': 'Some faces from this camera
cannot be processed',
'enabled': True,
'msg': '',
'statistic': { 'decoding_soft_errors': 0,
'faces_failed': 20,
'faces_not_posted': 0,
'faces_posted': 1051,
'frames_dropped': 4253,
'frame_height': 1080,
'frames_imotion_skipped': 0,
'frames_processed': 158401,
'frame_width': 1920,
'job_starts': 1,
'processed_duration': 6509.28,
'processing_fps': 48.428185},
'status': 'INPROGRESS'},
'id': 1,
'latitude': None,
'longitude': None,
'modified_date': '2020-07-20T06:38:58.607160Z',
'name': '1',
'screenshot': 'http://172.17.47.245/cameras/1/screenshot/',
'stream_settings': { 'api_timeout': 15000,
'draw_track': False,
'fd_frame_height': -1,
'ffmpeg_format': '',
'ffmpeg_params': [],
'jpeg_quality': 95,
'max_face_size': 0,
'md_scale': 0.3,
'md_threshold': 0.002,
'min_d_score': -1000,
'min_face_size': 0,
'min_score': -2,
'npersons': 4,
'overall': True,

```

(continues on next page)

(продолжение с предыдущей страницы)

```

        'realtime': False,
        'realtime_dly': 500,
        'realtime_post_perm': False,
        'roi': '',
        'rot': '',
        'tracker_threads': 4},
    'stream_settings_gpu': {'ffmpeg_format': '',
        'ffmpeg_params': [],
        'filter_max_face_size': 8192,
        'filter_min_face_size': 1,
        'filter_min_quality': 0.45,
        'imotion_threshold': 0,
        'jpeg_quality': 95,
        'overall_only': True,
        'play_speed': -1,
        'realtime_post_every_interval': False,
        'realtime_post_first_immediately': False,
        'realtime_post_interval': 1,
        'roi': '',
        'rot': '',
        'router_timeout_ms': 15000,
        'router_verify_ssl': True,
        'start_stream_timestamp': 0,
        'use_stream_timestamp': False},
    'threshold': None,
    'url': 'http://a3569458063-s26881.cdn.ngenix.net/live/smil:r24.smil/
↪ chunklist_b1200000.m3u8?codec=mpeg4'}],
    'closed_date': '2020-07-30T16:05:24.077331Z',
    'created_date': '2020-07-30T16:04:02Z',
    'event_type': 'episode_close',
    'events_count': 4,
    'features': None,
    'id': 104236,
    'last_event': {'acknowledged': True,
        'acknowledged_by': None,
        'acknowledged_date': '2020-07-30T16:05:09+00:00',
        'acknowledged_reaction': '',
        'camera': 1,
        'camera_group': 1,
        'confidence': 0.0,
        'created_date': '2020-07-30T16:05:06+00:00',
        'episode': 104236,
        'face': 'http://172.17.47.245/uploads/2020/07/30/event/160509_face_
↪ PXGmaZ.jpg',
        'features': {'age': None,
            'beard': None,
            'emotions': None,
            'gender': None,
            'glasses': None,
            'liveness': None,
            'medmask': None},
        'frame': 'http://172.17.47.245/uploads/2020/07/30/event/160509_full_

```

(continues on next page)

(продолжение с предыдущей страницы)

```

↪frame_YeIRKk.jpg',
  'frame_coords_bottom': 322,
  'frame_coords_left': 465,
  'frame_coords_right': 574,
  'frame_coords_top': 190,
  'id': '4284565716150084776',
  'looks_like_confidence': None,
  'matched': False,
  'matched_dossier': None,
  'matched_face': '',
  'matched_lists': [-1],
  'quality': -0.000468,
  'scores': {'liveness_score': None,
             'quality': -0.000468814512714,
             'track': {'first_timestamp': '2020-07-30T16:04:56',
                       'id': '313e117d86b3-1071',
                       'last_timestamp': '2020-07-30T16:05:08'},
             'track_duration_seconds': 8.159999999999854},
  'temperature': None},
'matched_event': None,
'matched_lists': [{'acknowledge': False,
                   'active': True,
                   'camera_groups': [],
                   'color': 'ffffff',
                   'comment': 'Default list for unmatched evenets',
                   'created_date': '2020-07-12T05:48:09.324264Z',
                   'id': -1,
                   'modified_date': '2020-07-12T05:48:09.324369Z',
                   'name': 'Unmatched',
                   'notify': False,
                   'permissions': {},
                   'threshold': None}],
'open': False,
'temperature': None,
'webhook_type': 'episodes']]

```

Запись счетчика (Verbose)

```

[{'camera': {'active': True,
            'azimuth': None,
            'comment': '',
            'created_date': '2020-07-12T05:57:50.459974Z',
            'group': 1,
            'health_status': {'code': 'green',
                              'code_desc': 'Всё в порядке',
                              'enabled': True,
                              'msg': '',
                              'statistic': {'decoding_soft_errors': 0,
                                             'faces_failed': 0,
                                             'faces_not_posted': 0,

```

(continues on next page)

(продолжение с предыдущей страницы)

```
'faces_posted': 55,
'frames_dropped': 309,
'frame_height': 1080,
'frames_imotion_skipped': 0,
'frames_processed': 3181,
'frame_width': 1920,
'job_starts': 3,
'processed_duration': 139.68,
'processing_fps': 184.37485},
'status': 'INPROGRESS'},
'id': 1,
'latitude': None,
'longitude': None,
'modified_date': '2020-07-30T19:56:40.773455Z',
'name': '1',
'screenshot': 'http://172.17.47.245/cameras/1/screenshot/',
'stream_settings': {'api_timeout': 15000,
'draw_track': False,
'fd_frame_height': -1,
'ffmpeg_format': '',
'ffmpeg_params': [],
'jpeg_quality': 95,
'max_face_size': 0,
'md_scale': 0.3,
'md_threshold': 0.002,
'min_d_score': -1000,
'min_face_size': 0,
'min_score': -2,
'npersons': 4,
'overall': True,
'realtime': False,
'realtime_dly': 500,
'realtime_post_perm': False,
'roi': '',
'rot': '',
'tracker_threads': 4},
'stream_settings_gpu': {'ffmpeg_format': '',
'ffmpeg_params': [],
'filter_max_face_size': 8192,
'filter_min_face_size': 1,
'filter_min_quality': 0.45,
'imotion_threshold': 0,
'jpeg_quality': 95,
'overall_only': False,
'play_speed': -1,
'realtime_post_every_interval': False,
'realtime_post_first_immediately': False,
'realtime_post_interval': 1,
'roi': '',
'rot': '',
'routert_timeout_ms': 15000,
'routert_verify_ssl': True,
```

(continues on next page)

(продолжение с предыдущей страницы)

```

        'start_stream_timestamp': 0,
        'use_stream_timestamp': False},
    'threshold': None,
    'url': 'http://a3569458063-s26881.cdn.ngenix.net/live/smil:r24.smil/
↪ chunklist_b1200000.m3u8?codec=mpeg4'},
    'camera_group': {'active': True,
                     'comment': '',
                     'created_date': '2020-07-12T05:48:09.537724Z',
                     'deduplicate': True,
                     'deduplicateDelay': 10,
                     'id': 1,
                     'labels': {},
                     'modified_date': '2020-07-17T01:41:22.944825Z',
                     'name': 'Default Camera Group',
                     'permissions': {'1': 'view', '2': 'view', '3': 'view'},
                     'threshold': None},
    'counter': {'active': True,
                'camera': 1,
                'count_interval': 5,
                'created_date': '2020-07-30T17:56:01.469273Z',
                'detect_faces': True,
                'detect_silhouettes': False,
                'id': 1,
                'modified_date': '2020-07-30T19:58:31.544631Z',
                'name': '123',
                'roi': [[0, 0], [1024, 0], [1024, 576], [0, 576]]},
    'counter_name': '123',
    'created_date': '2020-07-30T20:17:01.443532Z',
    'event_type': 'counter_record',
    'faces_bbox': [[[327, 289], [474, 289], [474, 485], [327, 485]]],
    'faces_count': 1,
    'fullframe': 'http://172.17.47.245/uploads/2020/07/30/counters/201701_fullframe_fDH9f4.
↪ jpg',
    'id': 1253,
    'silhouettes_bbox': None,
    'silhouettes_count': 0,
    'thumbnail': 'http://172.17.47.245/uploads/2020/07/30/counters/201701_thumb_I5fzIP.jpg
↪ ',
    'webhook_type': 'counters'}}]

```

3.3 Интеграции с партнерами

3.3.1 Genetec Security Center

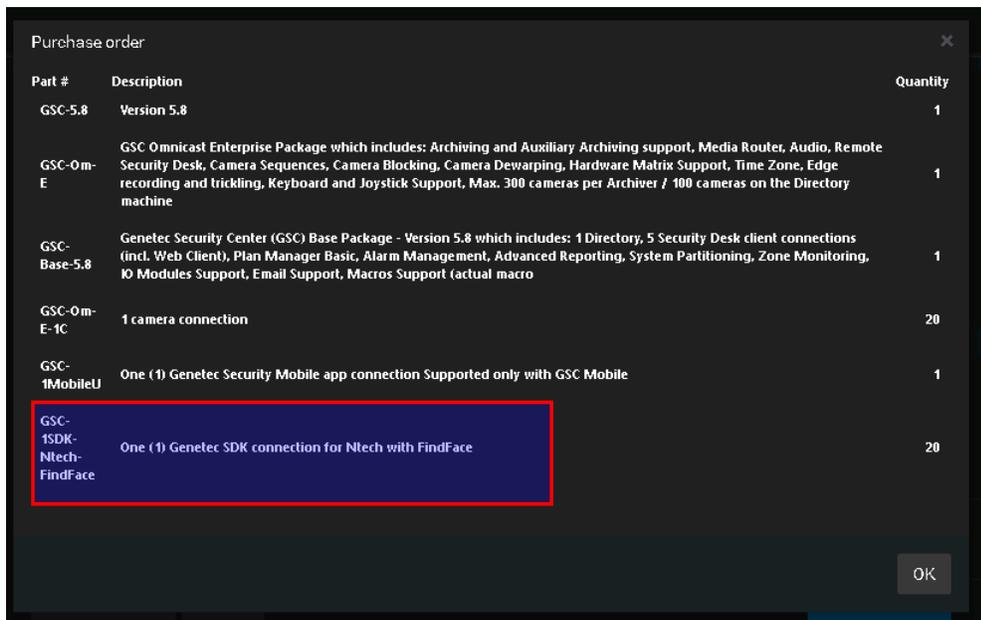
Интеграция ПК FindFace с программным комплексом Genetec Security Center позволяет добавлять функционал распознавания лиц в системы безопасности на базе Genetec.

Настройка интеграции

Интеграция с Genetec Security Center реализуется через плагин `findface-genetec`. По умолчанию плагин отключен.

Перед настройкой интеграции на стороне ПК FindFace разверните программное обеспечение Genetec Web SDK и Media Gateway и создайте в Genetec Security Center оповещение Alarm, которое будет отображаться при наступлении в ПК FindFace 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.



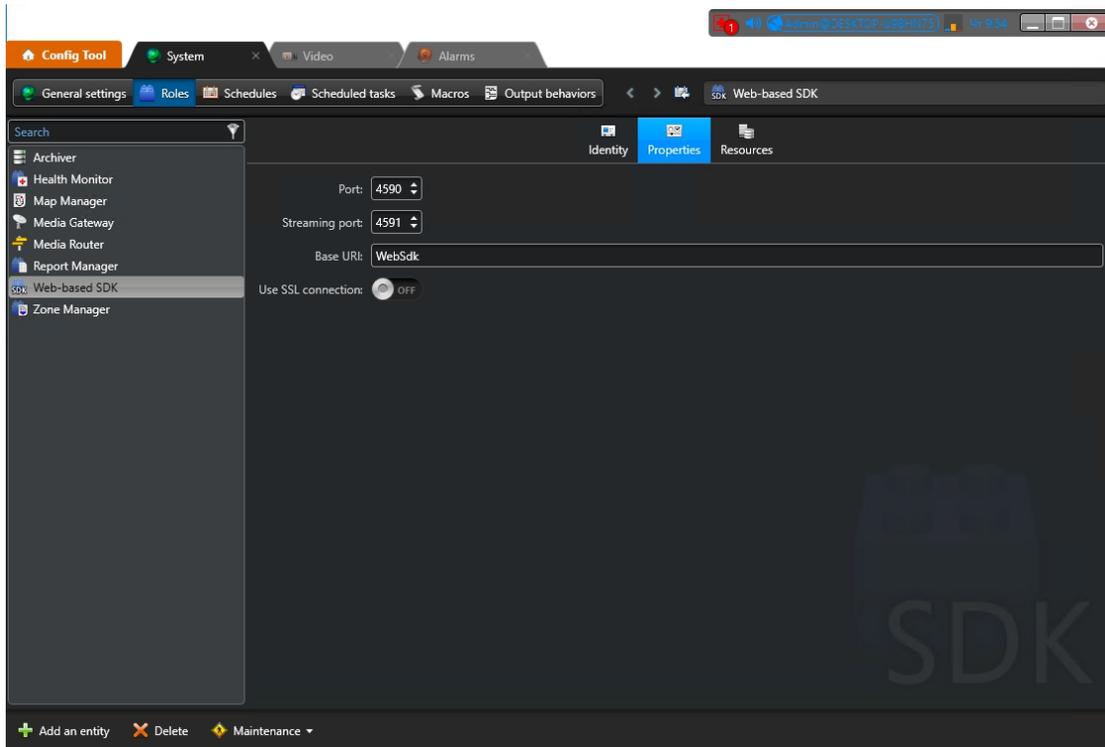
В этой главе:

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

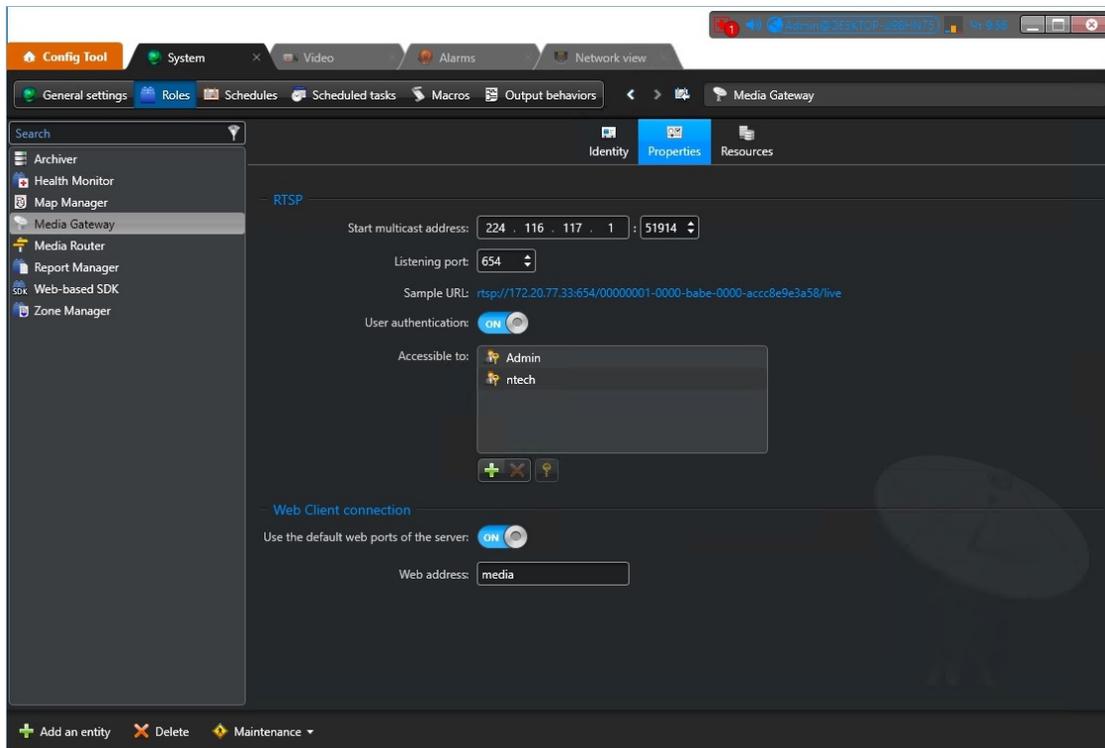
- Создание списков наблюдения и досье в ПК FindFace

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*.



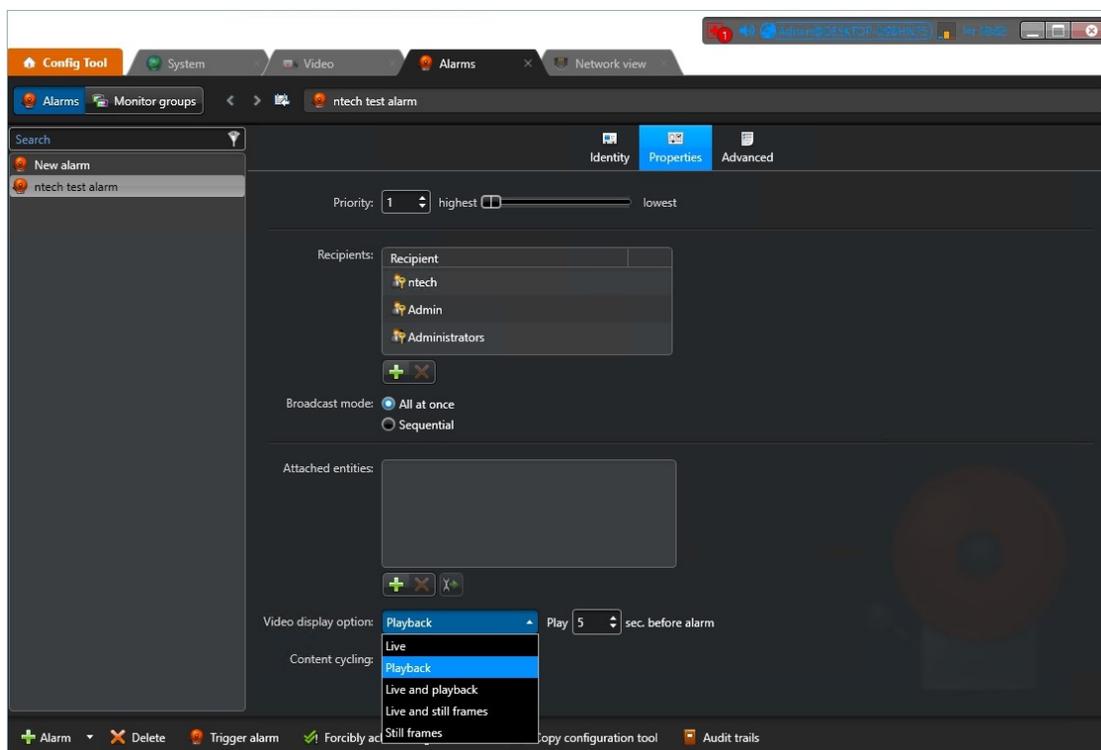
Для того чтобы развернуть Media Gateway в Genetec Config Tool, ознакомьтесь с содержанием главы *Security Center Administrator Guide* -> *Chapter 24: Video Deployment*.



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

Создание оповещения в Genetec Security Center

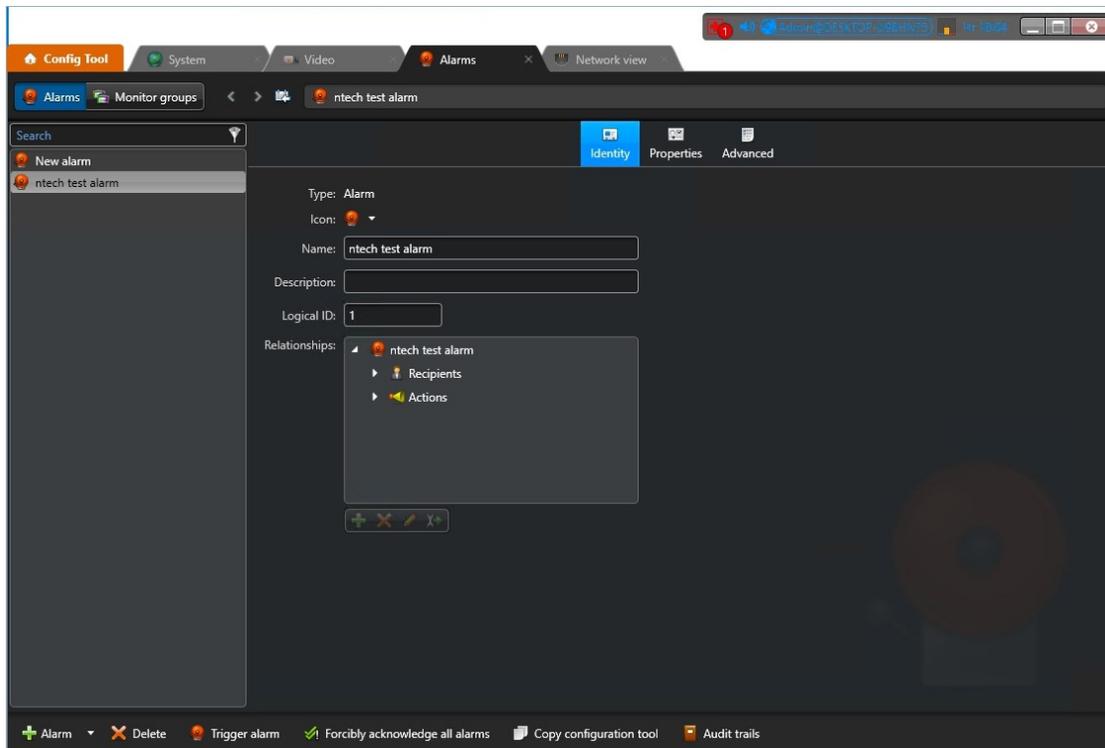
Создайте и настройте новое оповещение **Alarm** в Genetec Config Tool, руководствуясь разделом *Security Center Administrator Guide -> Chapter 48: Alarms -> Creating Alarms*.



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 ПК FindFace вам потребуется ввести логическое id оповещения, которое задается на вкладке *Identity*.



Включение интеграции Genetec в ПК FindFace

Для того чтобы активировать интеграцию Genetec в ПК FindFace, 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 ПК FindFace в PostgreSQL и заново создайте группы пользователей с *предустановленными* правами.

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

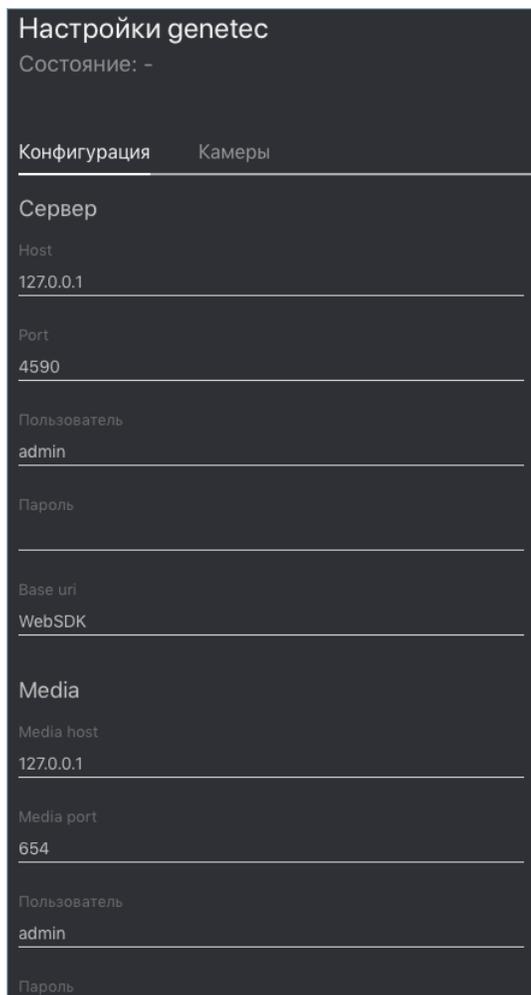
3. Перезапустите findface-security.

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

Настройка точек доступа в ПК FindFace

Для того чтобы установить соединение между ПК FindFace и Genetec Security Center, выполните следующие действия:

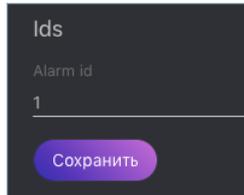
1. Перейдите на вкладку *Настройки*. Выберите *Genetec*.



2. В секциях *Сервер* и *Media*, укажите *настройки* точек доступа Web SDK и Media Gateway.

Важно: Порты WebSDK и Media Gateway должны быть открыты.

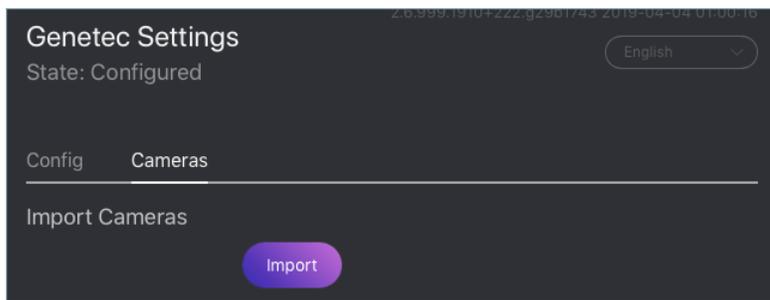
3. В секции `guiabel:Ids` укажите *логический id* оповещения Alarm, которое будет отображаться в Genetec Security Center при наступлении события распознавания лица в ПК FindFace.



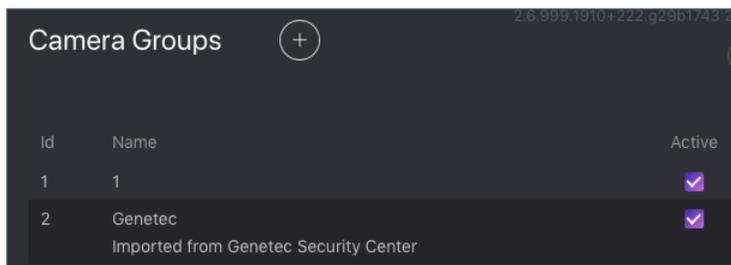
4. Нажмите *Сохранить*. Если соединение с Genetec Security Center успешно установлено, статус будет автоматически изменен на *Сконфигурирован*.

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.



Для того чтобы посмотреть список камер, на панели навигации ПК FindFace перейдите на вкладку *Камеры*. Для того чтобы исключить камеру из распознавания лиц, просто деактивируйте ее в этом списке.

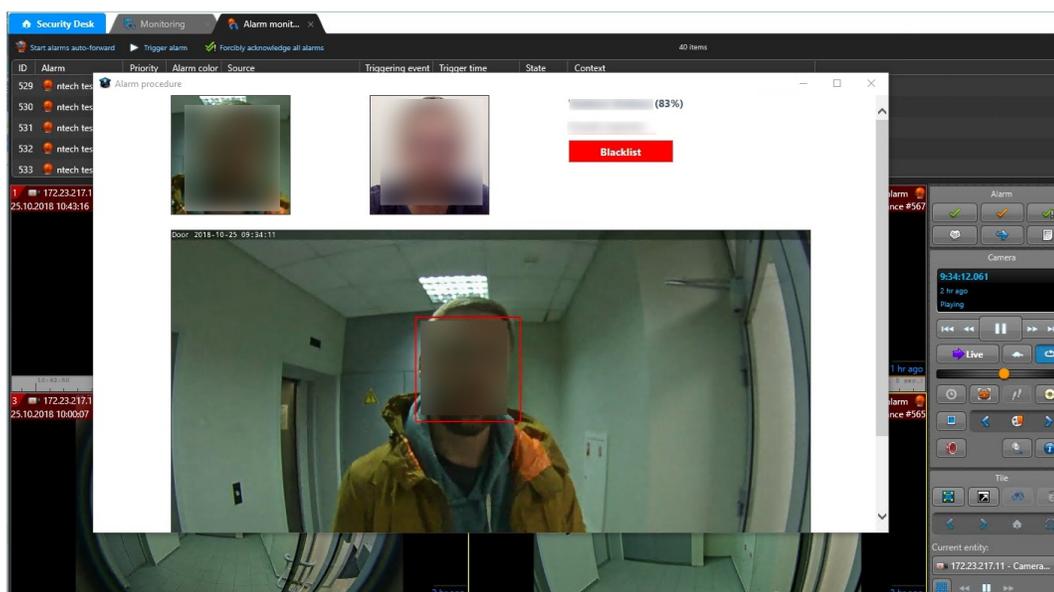
Создание списков наблюдения и досье в ПК FindFace

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 ПК FindFace, 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. ПК FindFace 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 Аххон Next

Интеграция ПК FindFace с программным комплексом Аххон Next позволяет обрабатывать видеопотоки из системы безопасности на базе Аххон и анализировать их на предмет наличия лиц из досье.

Важно: Один экземпляр ПК FindFace 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 ПК FindFace будет обращаться к API Аххон и за HLS-потоками архива. В параметре `rtsp` укажите общий адрес видеопотоков на сервере Аххон Next. `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 ПК FindFace (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 `аххон:vhod_1.Vhod_1`.

На этом настройка интеграции будет завершена. Если интеграция настроена корректно, ПК FindFace 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).

3.4 Пользовательские плагины

В ходе настройки системы вы можете установить собственные правила обработки обнаруженных на видео лиц. Для этого понадобится написать один или несколько плагинов на Python. Использование плагинов позволит вам выполнить настройку видеодетекции лиц индивидуально для каждого случая использования системы.

Для активации плагина используйте файл конфигурации `/etc/findface-facerouter.py`.

Совет: В качестве примера см. плагин по умолчанию `/opt/findface-security/fr_plugin/ffsec_fr_plugin.py`.

3.4.1 Развертывание findface-facerouter в ПК FindFace

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
```

3.4.2 Настройка `findface-facerouter` на использование плагинов

Совет: Для начала попробуйте плагин по умолчанию `/opt/findface-security/fr_plugin/ffsec_fr_plugin.py`. Или *создайте свой собственный*.

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 син-
таксически корректный код Python.
```

```
sudo vi /etc/findface-facerouter.py

plugin_dir = '/etc/findface/plugins/'
```

- Перезапустите `findface-facerouter`.

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

3.4.3 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.httptserver.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)
```

3.4.4 Классы и методы

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

n/a

`shutdown(self)`

Данный метод вызывается только перед завершением работы компонента `findface-facerouter`.

Параметры

n/п

Результат

n/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 экземпляра 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 <code>param</code> и <code>value</code> 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 <code>findface-extraction-api</code> .
SOURCE_NOT_FOUND	Параметре <code>from</code> задано несуществующее лицо.
SOURCE_GALLERY_NOT_FOUND	Параметру <code>from</code> 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`.

3.4.5 Пример

Следующий пример иллюстрирует основы написания плагина и использование классов и методов. Данный плагин запрашивает атрибуты лица у `findface-sf-api`, а затем отправляет запрос в `<FFSEC_URL>/video-detector/process` для создания события с данными, полученными из `findface-sf-api`.

Вы можете найти данный плагин в каталоге `/opt/findface-security/fr_plugin/ffsec_fr_plugin.py`. *Настройте* систему на его использование и попробуйте.

Важно: Убедитесь, что в переменной `FFSEC_URL` указаны актуальный IP-адрес и порт сервера `findface-security`.

```
import datetime
import logging
import aiohttp
from dateutil.tz import tzutc
from facerouter.plugin import Plugin
from ntech import sfapi_client
from ntech.asyncio_utils import wrap_futures
from ntech.asyncio_utils.noop_cookie import NoopCookieJar
from ntech.tornado_utils import asyncio_to_tornado
# change this if your ffsecurity is located on another host or listens on a non-default
↪port
FFSEC_URL = 'http://127.0.0.1:8002'
logger = logging.getLogger(__name__)
class FFSecurityPlugin(Plugin):
```

(continues on next page)

(продолжение с предыдущей страницы)

```

def __init__(self, ctx, ffsec_url):
    super().__init__(ctx)
    self.ffsec_url = ffsec_url.rstrip('/')
    self.session = aiohttp.ClientSession(cookie_jar=NoopCookieJar())
    self.future_wrapper = asyncio_to_tornado
def deactivate(self, *args):
    self.session.close()
def request_headers(self, request):
    return {
        "Authorization": request.headers['Authorization'],
        'X-Request-ID': request.request_id,
    }
@wrap_futures
async def preprocess(self, request, labels):
    # somewhat hacky way to pass data between preprocess and process:
    request.ffsec_reception_timestamp = datetime.datetime.now(tzutc())
    headers = self.request_headers(request)
    async with self.session.post(self.ffsec_url + '/video-detector/preprocess',
↳headers=headers) as resp:
        resp.raise_for_status()
        resp_json = await resp.json()
        logger.debug("request_id=%r preprocess: ffsecurity response: %r", request.
↳request_id, resp_json)
        plugin_wants = resp_json['plugin_wants']
        request.ffsec_plugin_wants = plugin_wants
        logger.info("request_id=%r preprocess: ffsecurity requested features: %r",
↳request.request_id, plugin_wants)
        return plugin_wants
@wrap_futures
async def process(self, request, photo, bbox, event_id, detection: sfapi_client.
↳DetectFace):
    headers = self.request_headers(request)
    with aiohttp.MultipartWriter('form-data') as mpwriter:
        part = aiohttp.payload.BytesPayload(request.params.photo)
        part.set_content_disposition('form-data', name='photo', filename='photo.jpg')
        mpwriter.append(part)
        part = aiohttp.payload.BytesPayload(b'')
        part.set_content_disposition('form-data', name='normalized', filename='norm.
↳png')
        mpwriter.append(part)
        part = aiohttp.payload.JsonPayload(request.params.detectorParams)
        part.set_content_disposition('form-data', name='detectorParams')
        mpwriter.append(part)
        part = aiohttp.payload.JsonPayload([list(bbox)])
        part.set_content_disposition('form-data', name='bbox')
        mpwriter.append(part)
        part = aiohttp.payload.StringPayload(request.params.cam_id)
        part.set_content_disposition('form-data', name='cam_id')
        mpwriter.append(part)
        part = aiohttp.payload.StringPayload(request.params.timestamp.isoformat())
        part.set_content_disposition('form-data', name='timestamp')
        mpwriter.append(part)

```

(continues on next page)

(продолжение с предыдущей страницы)

```

        part = aiohttp.payload.StringPayload(request.ffsec_reception_timestamp.
↪isoformat())
        part.set_content_disposition('form-data', name='reception_timestamp')
        mpwriter.append(part)
        part = aiohttp.payload.JsonPayload(request.ffsec_plugin_wants)
        part.set_content_disposition('form-data', name='plugin_wants')
        mpwriter.append(part)
        if request.params.bs_type is not None:
            part = aiohttp.payload.StringPayload(request.params.bs_type)
            part.set_content_disposition('form-data', name='bs_type')
            mpwriter.append(part)
        part = aiohttp.payload.JsonPayload({
            'id': getattr(detection, 'id', None),
            'features': detection.features,
            'bbox': detection.bbox._asdict(),
            'facen': getattr(detection, 'facen', None),
            'attributes': detection.attributes,
        })
        part.set_content_disposition('form-data', name='detection')
        mpwriter.append(part)
        async with self.session.post(
            self.ffsec_url + '/video-detector/process',
            data=mpwriter,
            headers=headers
        ) as resp:
            await resp.read()
            resp.raise_for_status()
        logger.info("request_id=%r process: ffsecurity accepted event", request.request_
↪id)
    ↪id)
async def activate(app, ctx, plugin_name, plugin_source):
    plugin = FFSecurityPlugin(ctx=ctx, ffsec_url=FFSEC_URL)
    return plugin

```


f

`facrouter.plugin`, 279

n

`ntech.sfapi_client.client`, 282

`ntech.sfapi_client.filters`, 286

`ntech.sfapi_client.gallery`, 283

O

`objects`, 281

СИМВОЛЫ

- `__init__()` (метод `ntech.sfapi_client.filters.Detection`), 288
- `__init__()` (метод `ntech.sfapi_client.filters.Face`), 288
- A**
- `add()` (метод `ntech.sfapi_client.gallery.Gallery`), 284
- B**
- `BBox` (класс в `objects`), 281
- C**
- `Client` (класс в `ntech.sfapi_client.client`), 282
- `create()` (метод `ntech.sfapi_client.gallery.Gallery`), 285
- D**
- `delete()` (метод `ntech.sfapi_client.gallery.Gallery`), 284
- `detect()` (метод `ntech.sfapi_client.client.Client`), 282
- `Detection` (класс в `ntech.sfapi_client.filters`), 288
- `drop()` (метод `ntech.sfapi_client.gallery.Gallery`), 285
- F**
- `Face` (класс в `ntech.sfapi_client.filters`), 288
- `facrouter.plugin` модуль, 279
- `Filter` (класс в `ntech.sfapi_client.filters`), 286
- G**
- `Gallery` (класс в `ntech.sfapi_client.gallery`), 283
- `gallery()` (метод `ntech.sfapi_client.client.Client`), 282
- `get()` (метод `ntech.sfapi_client.gallery.Gallery`), 284
- `gte()` (метод класса `ntech.sfapi_client.filters.Id`), 286
- `gte()` (метод класса `ntech.sfapi_client.filters.Meta`), 287
- I**
- `Id` (класс в `ntech.sfapi_client.filters`), 286
- L**
- `list()` (метод `ntech.sfapi_client.gallery.Gallery`), 283
- `lte()` (метод класса `ntech.sfapi_client.filters.Id`), 286
- `lte()` (метод класса `ntech.sfapi_client.filters.Meta`), 287
- M**
- `Meta` (класс в `ntech.sfapi_client.filters`), 286
- N**
- `ntech.sfapi_client.client` модуль, 282
- `ntech.sfapi_client.filters` модуль, 286
- `ntech.sfapi_client.gallery` модуль, 283
- O**
- `objects` модуль, 281
- `objects.DetectFace` (класс в `objects`), 281
- `objects.DetectResponse` (класс в `objects`), 281
- `objects.Face` (класс в `objects`), 281
- `objects.FaceId` (класс в `objects`), 281
- `objects.ListResponse` (класс в `objects`), 281
- `oneof()` (метод класса `ntech.sfapi_client.filters.Id`), 286
- `oneof()` (метод класса `ntech.sfapi_client.filters.Meta`), 287

P

Plugin (класс в *facrouter.plugin*), 279
preprocess(), 278
preprocess() (метод *facrouter.plugin.Plugin*), 279
process(), 279
process() (метод *facrouter.plugin.Plugin*), 280

S

serialize() (метод
 ntech.sfapi_client.filters.Filter), 286
sfapi_client.SFApiMalformedResponseError
 (класс в *ntech.sfapi_client.filters*), 289
sfapi_client.SFApiRemoteError (класс в
 ntech.sfapi_client.filters), 288
shutdown(), 279
shutdown() (метод *facrouter.plugin.Plugin*), 281
subset() (метод класса
 ntech.sfapi_client.filters.Meta), 287

U

update() (метод *ntech.sfapi_client.gallery.Gallery*),
 285

модуль

facrouter.plugin, 279
ntech.sfapi_client.client, 282
ntech.sfapi_client.filters, 286
ntech.sfapi_client.gallery, 283
objects, 281