

VisionLabs FaceStream

Руководство администратора

v.5.1.12

Содержание

Глоссарий	8
1 Введение	10
2 Аппаратные требования	11
2.1 Минимальные аппаратные требования	11
3 Программные требования	12
4 Обзор	13
4.1 Работа FaceStream с лицами и телами	15
4.1.1 Работа с лицами	15
4.1.2 Работа с телами	17
4.2 Взаимодействие FaceStream с LUNA Streams	19
4.2.1 Распределение потоков в LUNA Streams	21
4.2.1.1 Таблица переходов от статуса к статусу	22
4.2.2 Процесс обработки потоков в LUNA Streams	22
4.2.2.1 Автоматический перезапуск потоков	23
4.2.3 Группировка потоков	24
4.2.4 Описание базы данных LUNA Streams	24
4.3 Рекомендации по настройке FaceStream	26
4.3.1 Рекомендации перед началом настройки	26
4.3.2 Настройка производительности FaceStream	26
4.3.2.1 Уменьшение области поиска лиц	27
4.3.2.2 Масштабирование кадра	29
4.3.3 Определение области с движением	29
4.3.4 Обработка пакетов кадров	30
4.3.4.1 Минимальный размер лица	30
4.3.5 Общая информация о настройке	31
4.3.5.1 Работа с треком	31
4.3.5.2 Отправка лучшего кадра	31
4.3.5.3 Фильтрация кадров	32
4.3.5.4 Работа со СКУД	33
4.4 Форматы, стандарты сжатия видео и протоколы	35
4.4.1 Форматы видео	35
4.4.2 Стандарты сжатия видео	35
4.4.3 Протоколы	35
4.5 Потребление памяти при работе FaceStream	36

4.6	Мониторинг	37
4.6.1	InfluxDB OSS 2	37
4.6.1.1	Настройка InfluxDB	37
4.7	Интерфейс воспроизведения потока	39
5	Перечень приоритетных настроек	40
5.1	Настройки для работы с лицами	41
5.2	Настройки для работы с телами	42
6	Настройки управления потоками	44
6.1	Параметр «account_id»	44
6.2	Параметр «name»	44
6.3	Параметр «description»	45
6.4	Группа параметров «data»	46
6.4.1	Параметр «type»	46
6.4.2	Параметр «reference»	46
6.4.3	Параметр «roi»	47
6.4.4	Параметр «droi»	48
6.4.5	Параметр «rotation»	50
6.4.6	Параметр «preferred_program_stream_frame_width»	50
6.4.7	Параметр «endless»	51
6.4.8	Параметр «mask»	51
6.5	Группа параметров «event_handler»	53
6.5.1	Параметр «origin»	53
6.5.2	Параметр «api_version»	53
6.5.3	Параметр «handler_id» группы «bestshot_handler»	53
6.5.4	Параметр «handler_id» группы «detection_handler»	53
6.5.5	Параметр «frame_store»	54
6.5.6	Группа параметров «authorization»	55
6.6	Группа параметров «policies»	56
6.6.1	Группа параметров «sending»	56
6.6.1.1	Параметр «time_period_of_searching»	56
6.6.1.2	Параметр «silent_period»	57
6.6.1.3	Параметр «type»	57
6.6.1.4	Параметр «number_of_bestshots_to_send»	58
6.6.1.5	Параметр «send_only_full_set»	58
6.6.1.6	Параметр «delete_track_after_sending»	58
6.6.2	Группа параметров «primary_track_policy»	59
6.6.2.1	Параметр «use_primary_track_policy»	59
6.6.2.2	Параметр «best_shot_min_size»	59

6.6.2.3	Параметр «best_shot_proper_size»	59
6.6.3	Группа параметров «liveness»	61
6.6.3.1	Общие рекомендации по использованию Liveness	61
6.6.3.2	Параметр «use_shoulders_liveness_filtration»	65
6.6.3.3	Параметр «use_mask_liveness_filtration»	65
6.6.3.4	Параметр «use_flying_faces_liveness_filtration»	65
6.6.3.5	Параметр «liveness_mode»	65
6.6.3.6	Параметр «number_of_liveness_checks»	66
6.6.3.7	Параметр «liveness_threshold»	66
6.6.3.8	Параметр «liveness_weights»	66
6.6.3.9	Параметр «mask_backgrounds_count»	66
6.6.4	Группа параметров «filtering»	68
6.6.4.1	Параметр «min_score»	68
6.6.4.2	Параметр «detection_yaw_threshold»	68
6.6.4.3	Параметр «detection_pitch_threshold»	69
6.6.4.4	Параметр «detection_roll_threshold»	69
6.6.4.5	Параметр «yaw_number»	69
6.6.4.6	Параметр «yaw_collection_mode»	70
6.6.4.7	Параметр «mouth_occlusion_threshold»	70
6.6.5	Параметр «frame_processing_mode»	71
6.6.6	Параметр «real_time_mode_fps»	71
6.6.7	Параметр «ffmpeg_threads_number»	72
6.6.8	Группа параметров «health_check»	73
6.6.8.1	Параметр «max_error_count»	73
6.6.8.2	Параметр «period»	73
6.6.8.3	Параметр «retry_delay»	73
6.7	Группа параметров «location»	74
6.8	Группа параметров «autorestart»	74
6.9	Параметр «status»	75
6.10	Параметры «group_name» и «group_id»	75

7 Настройки, задаваемые в сервисе LUNA Configurator 76

7.1	Настройки FaceStream	76
7.1.1	Группа параметров «logging»	76
7.1.1.1	Параметр «severity»	76
7.1.1.2	Параметр «tags»	77
7.1.1.3	Параметр «mode»	78
7.1.2	Группа параметров «sending»	78
7.1.2.1	Параметр «request_type»	78
7.1.2.2	Параметр «portrait_type»	79

7.1.2.3	Параметр «send_source_frame»	80
7.1.2.4	Параметр «size_source_frame»	81
7.1.2.5	Параметр «detection_path_length»	81
7.1.2.6	Параметр «minimal_body_track_length_to_send»	81
7.1.2.7	Параметр «async_requests»	83
7.1.2.8	Параметр «aggregate_attr_requests»	83
7.1.2.9	Параметр «jpeg_quality_level»	84
7.1.3	Группа параметров «lunastreams»	84
7.1.3.1	Параметр «origin»	84
7.1.3.2	Параметр «api_version»	84
7.1.3.3	Параметр «max_number_streams»	85
7.1.3.4	Параметр «request_stream_period»	85
7.1.3.5	Параметр «send_feedback_period»	85
7.1.3.6	Параметр «max_feedback_delay»	86
7.1.4	Группа параметров «performance»	86
7.1.4.1	Параметр «stream_images_buffer_max_size»	86
7.1.4.2	Параметр «enable_gpu_processing»	86
7.1.4.3	Параметр «convert_full_frame»	87
7.1.5	Группа параметров «debug»	87
7.1.5.1	Параметр «save_debug_info»	87
7.1.5.2	Параметр «save_only_jpegs_with_honest_detections»	88
7.1.5.3	Параметр «save_jpegs»	88
7.2	Настройки TrackEngine	89
7.2.1	Параметры «use-face-detector» и «use-body-detector»	89
7.2.2	Параметр «detector-step»	89
7.2.3	Параметр «detector-scaling»	89
7.2.4	Параметр «scale-result-size»	90
7.2.5	Параметр «skip-frames»	92
7.2.6	Параметр «frg-subtractor»	93
7.2.7	Параметр «frg-regions-alignment»	93
7.2.8	Параметр «frg-regions-square-alignment»	93
7.2.9	Параметр «batched-processing»	94
7.2.10	Параметр «min-frames-batch-size»	94
7.2.11	Параметр «max-frames-batch-gather-timeout»	94
7.3	Настройки LUNA Streams	96
7.3.1	Группа параметров «LUNA_STREAMS_DB»	96
7.3.1.1	Параметр «db_type»	96
7.3.1.2	Параметр «db_user»	96
7.3.1.3	Параметр «db_password»	96

7.3.1.4	Параметр «db_name»	96
7.3.1.5	Параметр «db_host»	97
7.3.1.6	Параметр «db_port»	97
7.3.1.7	Параметр «connection_pool_size»	97
7.3.2	Группа параметров «LUNA_STREAMS_LOGGER»	97
7.3.2.1	Параметр «log_level»	97
7.3.2.2	Параметр «folder_with_logs»	98
7.3.2.3	Параметр «log_time»	98
7.3.2.4	Параметр «log_to_stdout»	98
7.3.2.5	Параметр «log_to_file»	98
7.3.2.6	Параметр «multiline_stack_trace»	98
7.3.3	Группа параметров «LUNA_LICENSES_ADDRESS»	99
7.3.3.1	Параметр «origin»	99
7.3.3.2	Параметр «api_version»	99
7.3.4	Параметр «STREAM_WORKER_ASYNC_LOCK_TIMEOUT»	99
7.3.5	Параметр «STREAM_STATUS_OBSOLETING_PERIOD»	99
7.3.6	Параметр «LUNA_STREAMS_ACTIVE_PLUGINS»	99
7.3.7	Параметр «STORAGE_TIME»	100
7.3.8	Группа параметров «INFLUX_MONITORING»	100
7.3.8.1	Параметр «send_data_for_monitoring»	100
7.3.8.2	Параметр «use_ssl»	100
7.3.8.3	Параметр «flushing_period»	100
7.3.8.4	Параметр «host»	100
7.3.8.5	Параметр «port»	101
7.3.8.6	Параметр «bucket»	101
7.3.8.7	Параметр «organization»	101
7.3.8.8	Параметр «token»	101
8	Использование FaceStream с LUNA Configurator	102
8.1	Особенности работы с LUNA Configurator	102
8.2	Параметры в Configurator	102
8.3	Задание настроек для нескольких экземпляров FaceStream	103
9	Использование FaceStream с конфигурационными файлами	105
9.1	Ключи запуска с конфигурационными файлами	106
10	Вывод информации в логи	108
10.1	Формат вывода логов FaceStream	108
10.2	Ошибки сервиса LUNA Streams	108
10.2.1	Вернулся код ошибки 39001	109

10.2.2	Вернулся код ошибки 39002	109
10.2.3	Вернулся код ошибки 39003	109
10.2.4	Вернулся код ошибки 39004	110
10.2.5	Вернулся код ошибки 39005	110
10.2.6	Вернулся код ошибки 39006	110
10.2.7	Вернулся код ошибки 39007	111
10.2.8	Вернулся код ошибки 39008	111
10.2.9	Вернулся код ошибки 39009	111
10.2.10	Вернулся код ошибки 39010	112

11 Совместимость с камерами

113

Глоссарий

Термин	Значение термина
Биометрический образец	Изображения, содержащие лицо или тело и соответствующие стандарту VisionLabs. Используется при работе с LUNA PLATFORM.
Биометрический шаблон	Набор уникальных свойств, получаемых в LUNA PLATFORM из биометрического образца.
Детекция	Сущность FaceStream, содержащая координаты лица или тела и оценочное значение объекта, по которому определяется лучший кадр.
Лучший кадр	Лучший кадр выбирается из всех кадров одного трека. Основными условиями выбора лучшего кадра являются приемлемое качество изображения и наличие на нём лица или тела с наилучшим ракурсом. Условия выбора лучшего кадра задаются в настройках FaceStream.
Портрет	Изображение лица или тела, трансформированное под определенный формат. Портрет имеет два типа - «warp» (изображение трансформируется в формат биометрического образца), «gost» (из исходного кадра вырезается детекция с учетом отступов).
Ракурс	Степень поворота головы (в градусах) по каждой из трех осей вращения (наклон вверх/вниз относительно горизонтальной оси; наклон влево/вправо относительно вертикальной оси; поворот относительно вертикальной оси).
Событие	Сущность LUNA PLATFORM, которая содержит информацию (город, пользовательские данные, номер трека и т.д.) об одном лице и/или теле. Данная информация передается в LUNA PLATFORM приложением FaceStream. Полный перечень передаваемой информации см. в документации OpenAPI LUNA PLATFORM.
Трек	Информация о положении объекта (лица) одного человека на последовательности кадров. Если объект покидает зону кадра, то трек прерывается не сразу. Некоторое время он ожидает возвращения объекта в кадр. Если объект вернулся, то трек продолжается.

Термин	Значение термина
Трекинг	Функция отслеживания объекта (лица) на последовательности кадров.

1 Введение

Данный документ описывает:

- аппаратные и программные требования
- общее описание работы приложения и рекомендации по настройке
- процесс взаимодействия с LUNA Streams
- перечень основных настроек, необходимых для запуска FaceStream
- детальное описание всех настроек
- использование FaceStream с LUNA Configurator
- использование FaceStream с конфигурационными файлами
- ошибки API для FaceStream и LUNA Streams
- информацию о совместимости с моделями камер

Подробную информацию по запуску приложения см. в руководстве по установке FaceStream.

2 Аппаратные требования

2.1 Минимальные аппаратные требования

Дальнейшие минимальные требования приведены для использования одного экземпляра FaceStream.

Для корректной работы приложения аппаратное обеспечение должно отвечать следующим минимальным требованиям:

- CPU с частотой 2 ГГц и выше;
- 4 Гб оперативной памяти и выше;
- 10 Гб свободного места на жестком диске.
- Доступ к Интернету (для контейнеров и дополнительных загрузок ПО).

На аппаратные требования влияют несколько факторов:

- Количество обрабатываемых потоков;
- Частота и разрешение кадров потоков;
- Параметры настройки FaceStream. Настройки по умолчанию являются наиболее универсальными. В зависимости от условий эксплуатации приложения с помощью их значений можно повлиять на качество или производительность.

Следует подбирать аппаратное обеспечение на основе вышеперечисленных факторов.

FaceStream также может работать в режиме ускорения вычислений за счет:

Использования ресурсов видеокарты

Вычисления с использованием видеокарты поддерживаются только для детектора FaceDetV3. См. параметр «defaultDetectorType» в настройках FaceEngine («faceengine.conf»).

Требуется минимум 6Гб оперативной или выделенной видеопамяти. Рекомендуется 8 Гб VRAM или более.

Поддерживаются архитектуры Pascal, Volta, Turing. Требуется Compute Capability 6.1 или выше и CUDA версии 11.4.

Рекомендуемый драйвер NVIDIA - 470.103.01.

В данный момент для одного экземпляра FaceStream поддерживается только одна видеокарта.

Использования AVX2 инструкций

Требуется CPU с поддержкой AVX2. Система автоматически определяет наличие инструкций и запускается в оптимальном режиме.

3 Программные требования

Запуск контейнеров FaceStream и LUNA Streams был протестирован на:

- CentOS Linux release 7.8.2003 (Core)

В контейнере FaceStream используется для запуска следующая ОС:

- CentOS Linux release 8.3.2011

Для запуска контейнеров должен быть установлен Docker. Для загрузки настроек в сервис LUNA Configurator требуется наличие Python версии 2.x или 3.x.

4 Обзор

Приложение FaceStream осуществляет несколько функций:

- **Чтение потоков**

В качестве источников данных могут выступать веб-камеры, USB и IP-камеры (посредством протокола RTSP), видеофайлы и фотоизображения.

- **Обработка потоков**

Выполняется поиск и сопровождение лиц или тел в потоке, пока они не покинут кадр или не будут перекрыты.

- **Отсылка изображений лиц или тел в виде HTTP-запросов во внешний сервис**

В качестве внешнего сервиса выступает ПО компании VisionLabs LUNA PLATFORM 5.

Работа FaceStream зависит от пяти настроек.

- [Настройки управления потоками](#), задаваемые в LUNA Streams.

Здесь задаются настройки, касающиеся источников потоков, такие как тип источника, адрес источника, настройки фильтрации и др. Настройки задаются с помощью отправки запросов с телом в формате JSON в сервис LUNA Streams. FaceStream забирает настройки из LUNA Streams для дальнейшей обработки. Подробное описание работы FaceStream с LUNA Streams приведено в разделе [«Взаимодействие FaceStream с LUNA Streams»](#).

- [Настройки FaceStream](#), задаваемые в LUNA Configurator.

Здесь задаются общие настройки FaceStream, такие как логирование, настройка отправки готовых изображений из FaceStream во внешние сервисы, отладка и т.п.

- [Настройки TrackEngine](#), задаваемые в LUNA Configurator.

Здесь задаются настройки, касающиеся детекции и трекинга лица или тела.

- [Настройки LUNA Streams](#), задаваемые в LUNA Configurator.

Здесь задаются общие настройки сервиса LUNA Streams, такие как логирование, настройки БД, адрес сервиса LUNA Licenses и т.п.

- Настройки FaceEngine, задаваемые в конфигурационном файле «faceengine.conf» и передаваемые во время запуска контейнера FaceStream.

Здесь задаются параметры, касающиеся глубокой настройки распознавания лица. Изменять параметры рекомендуется только по согласованию с сотрудниками VisionLabs.

При работе с FaceStream также доступны следующие возможности:

- Динамическое создание, изменение, удаление источников потоков посредством запросов API

- Просмотр потоков в браузере с указанными параметрами в режиме реального времени
- Получение метрик по потоку (количество потоков, количество ошибок, количество необработанных кадров, FPS)

FaceStream можно настроить на работу либо с лицами, либо с телами. Одновременная обработка лиц и тел невозможна.

4.1 Работа FaceStream с лицами и телами

FaceStream может обрабатывать как лица, так и тела. Для каждого объекта предусмотрена своя схема работы и свой набор параметров, описанных ниже.

Необходимый минимум параметров для работы с обоими объектами можно найти в разделе «Перечень приоритетных настроек».

4.1.1 Работа с лицами

Схема работы FaceStream с лицами представлена на рисунке ниже.

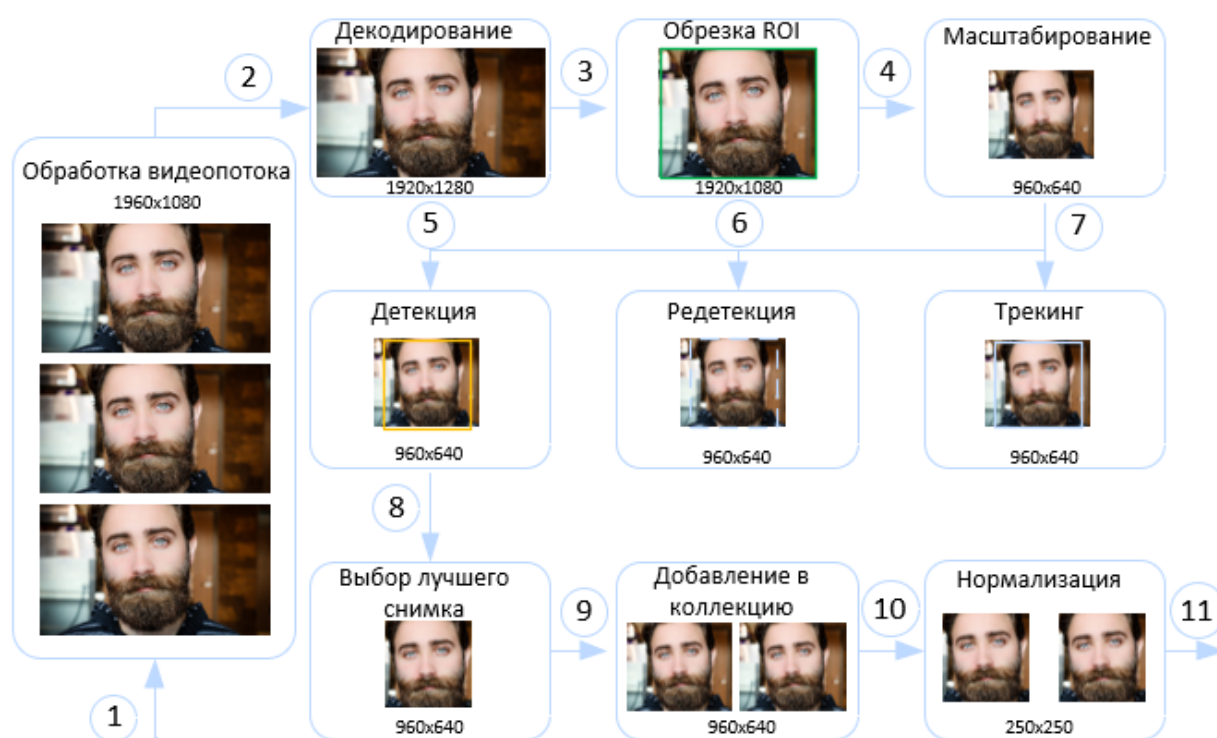


Рис. 1: Схема работы FaceStream с лицами

1. В FaceStream отправляется видео с камеры видеонаблюдения, веб камеры, видеофайл или фотоизображения. FaceStream может работать сразу с несколькими источниками потока (количество задается лицензией). Источники задаются с помощью отправки запросов с нужными параметрами в сервис [LUNA Streams](#);
2. FaceStream выполняет декодирование кадров видео;
3. Из кадра вырезается зона ROI, если задан параметр «roi» (настройки управления потоками);
4. Полученное изображение масштабируется до размера «scale-result-size» (настройки TrackEngine) если задан параметр «detector-scaling» (настройки TrackEngine);

5. Выполняется детекция лиц на кадре;
6. Вместо детекции может выполняться редетекция лица в кадре, если задан параметр `«detector-step»` (настройки TrackEngine);
7. Для каждого нового лица в потоке создается трек, который подкрепляется новыми детекциями данного лица с последующих кадров. Трек прерывается, если лицо исчезает из кадра. Можно задать параметр `«skip-frames»` (настройки TrackEngine), чтобы трек не прерывался сразу. Система будет ждать появления лица в области указанное количество кадров;
8. FaceStream отсекает кадры плохого качества и выбирает лучшие снимки. Существует несколько алгоритмов выбора лучшей детекции/детекций в треке. См. описание группы параметров `«filtering»` (настройки управления потоками);
9. Если кадр является лучшим снимком, он добавляется в коллекцию лучших снимков. В зависимости от настройки `«number_of_bestshots_to_send»` (настройки управления потоками) для каждого трека выбирается лучшая детекция в треке или несколько детекций;
10. **Опционально.** Если в настройке `«portrait_type»` (настройки управления потоками) указан параметр `«warp»`, то лучшие снимки приводятся к формату LUNA PLATFORM и создаются биометрические образцы. Биометрический образец лучше подходит для обработки с помощью LUNA PLATFORM;
11. Лучшие снимки отправляются во внешний сервис в виде HTTP запроса. Во внешнем сервисе могут генерироваться события по заданным обработчикам (см. описание события в руководстве администратора LUNA PLATFORM). Лучшие снимки могут быть переданы в исходном виде или трансформированы в биометрические образцы.

Частота отправки кадров задаётся в секции `«sending»` (настройки управления потоками).

Параметры отправки и адрес внешнего сервиса задаются в разделах `«data»` (настройки управления потоками) и `«sending»` (настройки FaceStream).

4.1.2 Работа с телами

Схема работы FaceStream с телами представлена на рисунке ниже.

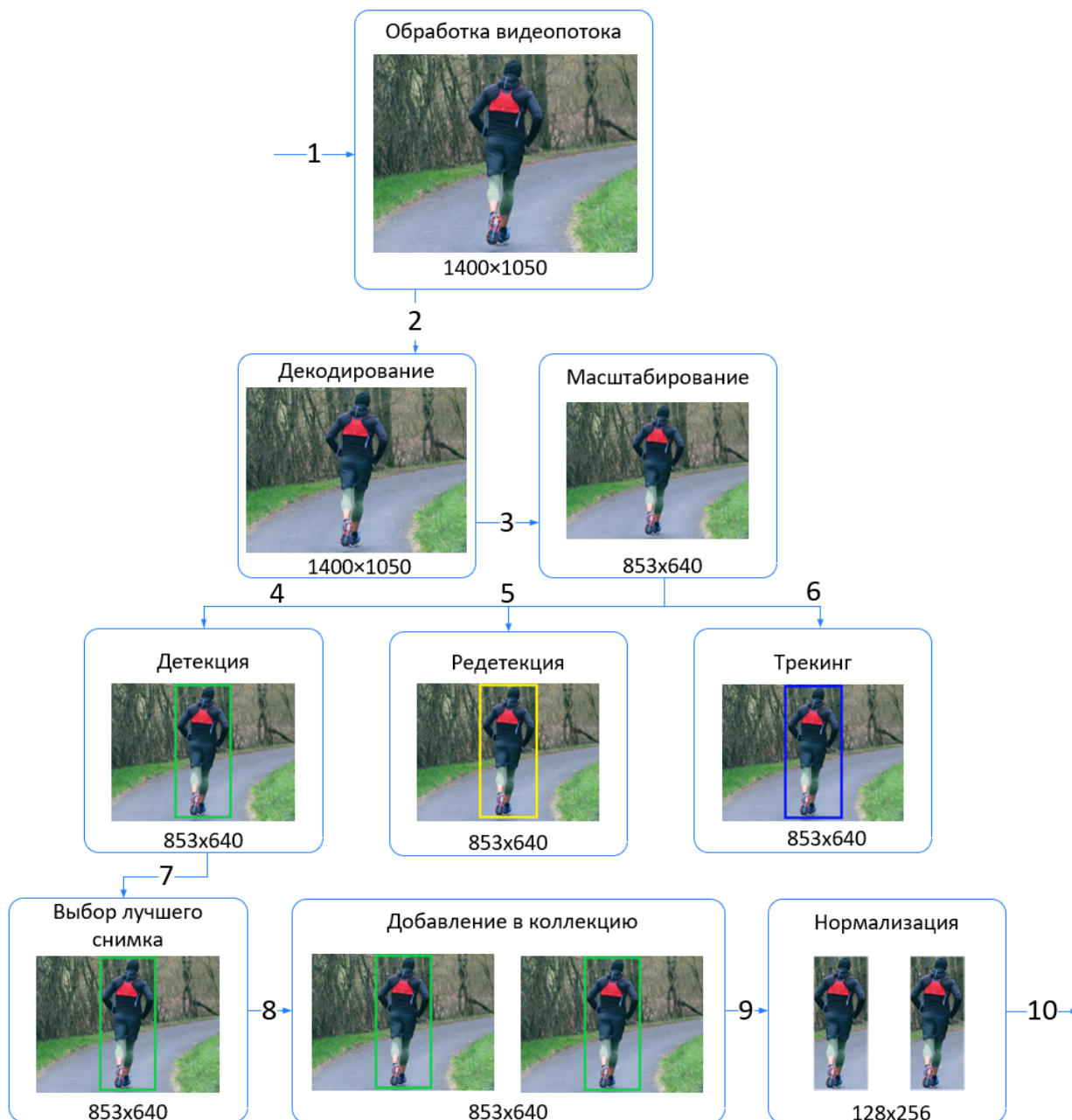


Рис. 2: Схема работы FaceStream с телами

1. В FaceStream отправляется видео с камеры видеонаблюдения, веб камеры, видеофайл или фотоизображения. FaceStream может работать сразу с несколькими источниками потока. Источники задаются с помощью отправки запросов с нужными параметрами в сервис [«LUNA Streams»](#);

2. FaceStream выполняет декодирование кадров видео;
3. Полученное изображение масштабируется до размера «[scale-result-size](#)» (настройки TrackEngine) если задан параметр «[detector-scaling](#)» (настройки TrackEngine);
4. Выполняется детекция тел на кадре;
5. Вместо детекции может выполняться редетекция тела в кадре, если задан параметр «[detector-step](#)» (настройки TrackEngine);
6. Для каждого нового тела в потоке создается трек, который подкрепляется новыми детекциями данного тела с последующих кадров. Трек прерывается, если тело исчезает из кадра. Можно задать параметр «[skip-frames](#)» (настройки TrackEngine), чтобы трек не прерывался сразу. Система будет ждать появления тела в области указанное количество кадров;
7. FaceStream отсекает кадры плохого качества и выбирает лучшие снимки. См. описание параметра «[min-score](#)» (настройки управления потоками);
8. Если кадр является лучшим снимком, он добавляется в коллекцию лучших снимков. В зависимости от параметра «[number_of_bestshots_to_send](#)» (настройки управления потоками) для каждого трека выбирается лучшая детекция в треке или несколько детекций;
9. Лучшие снимки приводятся к формату LUNA PLATFORM и создаются биометрические образцы. Биометрический образец лучше подходит для обработки с помощью LUNA PLATFORM;
10. Лучшие снимки отправляются во внешний сервис в виде HTTP запроса. Во внешнем сервисе могут генерироваться события по заданным обработчикам (см. описание события в руководстве администратора LUNA PLATFORM). Лучшие снимки трансформируются в биометрические образцы. Вместе с лучшими снимками могут отправляться координаты тела человека, если их количество задано в параметре «[minimal_body_track_length_to_send](#)» (настройки управления потоками).

Частота отправки кадров задаётся в секции «[sending](#)» (настройки управления потоками).

Параметры отправки и адрес внешнего сервиса задаются в разделах «[data](#)» (настройки управления потоками) и «[sending](#)» (настройки FaceStream).

4.2 Взаимодействие FaceStream с LUNA Streams

Для работы FaceStream необходимо предварительно запускать дополнительный сервис - LUNA Streams (порт по умолчанию - 5160). В теле запроса к сервису LUNA Streams задаются [настройки для управления потоками](#). После отправки запроса создается поток, чьи настройки забирает FaceStream для дальнейшей обработки. Примеры запросов можно найти в [спецификации Open API LUNA Streams](#).

Для использования сервиса LUNA Streams необходимо использовать сервисы LUNA PLATFORM 5 - LUNA Licenses и LUNA Configurator, а также базы данных PostgreSQL или Oracle и Influx.

База данных Influx нужна для целей [мониторинга](#) состояния сервисов LUNA PLATFORM. При необходимости мониторинг может быть отключен.

В документации FaceStream не описывается использование базы данных Oracle.

При необходимости можно запустить LUNA Streams без LUNA Configurator. Данный способ не описывается в документации.

FaceStream лицензируется с помощью ключа LUNA PLATFORM 5, в котором содержится информация о максимальном количестве потоков, которые может обработать LUNA Streams. Лицензия регистрируется сервисом LUNA Licenses.

См. подробную информацию по активации лицензии LUNA Streams в руководстве по установке FaceStream.

В базе данных PostgreSQL/Oracle хранятся все данные потоков LUNA Streams.

Общий процесс взаимодействия FaceStream с LUNA Streams представлен ниже:

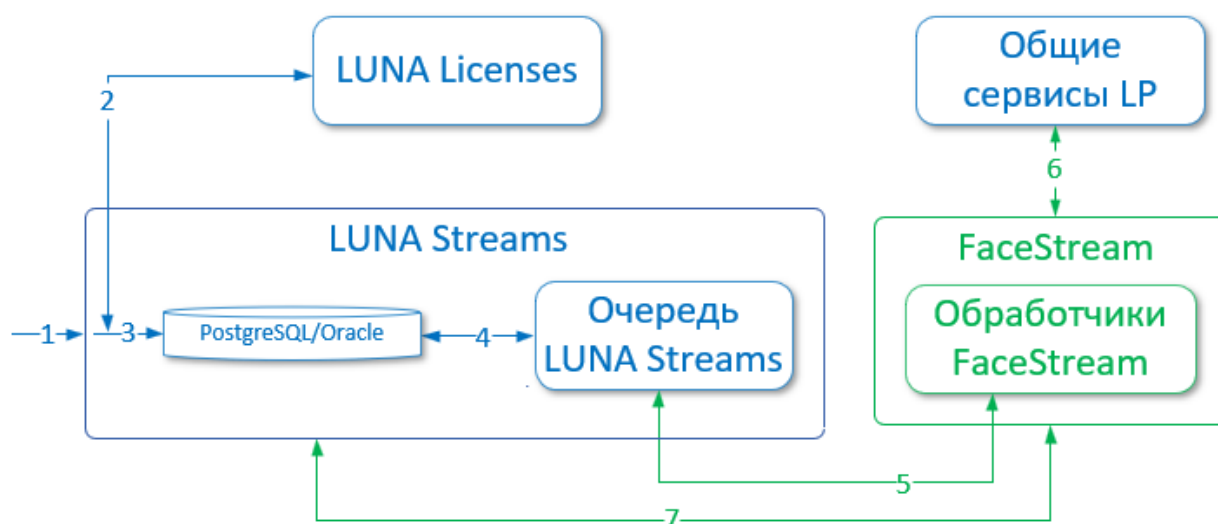


Рис. 3: Взаимодействие FaceStream с LUNA Streams

После отправки HTTP-запроса с заданными параметрами к сервису LUNA Streams **(1)**, выполняется проверка наличия параметра, регулирующего количество потоков для работы LUNA Streams, у ключа LUNA PLATFORM, с помощью сервиса LUNA Licenses **(2)**. Также проверяется количество уже обрабатываемых потоков на момент запроса с помощью отчета FaceStream **(7)** (см. ниже).

Если параметр ключа отсутствует, будет выдана ошибка лицензии.

Если на момент создания потока еще не обрабатывается максимальное количество доступных потоков, параметры добавляются в базу данных LUNA Streams **(3)** под уникальным идентификатором `stream_id`. Поток с параметрами попадает в очередь **(4)**, где находится в статусе «pending» пока специальный обработчик FaceStream не заберёт поток из очереди для последующей обработки.

Посмотреть потоки в очереди, отфильтровав их определенным образом, можно с помощью GET-запроса [«streams/processing/queue»](#).

Если на момент создания потока уже обрабатывается максимальное количество, то LUNA Streams не сможет добавить параметры в базу данных и будет выдана ошибка лицензии.

Если на момент создания потока FaceStream отключен, то может быть создано только то количество потоков со статусом «pending», которое оговорено лицензией. После запуска FaceStream будут приняты в обработку потоки, созданные в порядке очереди.

Потоки можно создавать со статусом «pause». В таком случае они будут добавлены в базу данных и будут ожидать ручного обновления статуса до «pending».

Очередь реализована в самом сервисе LUNA Streams и не является внешней.

Далее обработчики FaceStream забирают параметры потока/потоков из очереди **(5)** со статусом «pending» и начинают обработку. При этом статус обрабатываемых потоков меняется на «in_progress» и поток удаляется из очереди.

В процессе обработки регулярно отправляются данные в основные сервисы LUNA PLATFORM 5 для дальнейшей обработки кадров по заданным `handler_id` и для создания событий **(6)**, а также регулярно отправляется отчёт об обработке потоков в LUNA Streams **(7)**.

Время отправки отчетов является фиксированным и не может быть изменено.

Если в отчёте сказано, что какой-то поток был обработан, то обработчик FaceStream забирает следующие параметры потока со статусом «pending» из очереди LUNA Streams **(5)**, и сервис меняет статус потока с «pending» на «in_progress», удаляя из очереди. Если по неизвестным причинам отчет не был передан, то потоки заново помещаются в очередь.

Более подробное описание обработки потока LUNA Streams см. в разделе [«Процесс обработки потоков в LUNA Streams»](#).

Настройки для управления потоками задаются с помощью POST запроса к ресурсу [«/streams»](#).

Кроме того, для потока доступны следующие действия:

- получение существующих потоков по их «stream_id» с описанием данных каждого потока (запрос «[get streams](#)»)
- получение всей информации по потоку по его «stream_id», в т.ч. габаритах и частоте кадров, битрейту, группе кадров (gor), времени создания и начала обработки потока, последней ошибке при обработке и др. (запрос «[get stream](#)»)
- удаление существующих потоков по их «stream_id» (запрос «[delete streams](#)»)
- удаление потока по его «stream_id» (запрос «[remove stream](#)»)
- получение числа созданных потоков (запрос «[count streams](#)»)
- обновление полей «description» и «status» потока по его «stream_id» (запрос «[update stream](#)»)
- замена всех данных потока на новые по его «stream_id» (запрос «[put stream](#)»)
- получение ссылки на последний кадр (запрос «[get last frame preview](#)»)
- получение ссылки на поток, запущенный в реальном времени (запрос «[get live preview](#)»)
- получение логов потока (запрос «[get streams logs](#)»)
- удаление логов потока (запрос «[delete streams logs](#)»)

Подробное описание запросов и примеры запросов можно найти в [спецификации Open API LUNA Streams](#).

4.2.1 Распределение потоков в LUNA Streams

Как было сказано ранее, доступна возможность обработки нескольких потоков одновременно.

Для каждого потока предполагается его текущий статус:

- pending - поток взят в работу, но пока не найдено ни одного обработчика
- in_progress - поток обрабатывается
- done - поток полностью обработан (актуально для видеофайлов)
- pause - пользователь остановил обработку потока (не применимо для видеофайлов)
- restart - сервер перезапустил обработку потока
- cancel - пользователь отменил обработку потока (актуально для видеофайлов, но может применяться и для других источников)
- failure - при обработке потока произошла ошибка
- handler_lost - обработчик FaceStream потерян, следует передать другому (не применимо для видеофайлов)
- not_found - поток был в обработке, но в процессе выполнения был удалён
- deleted - намеренное удаление потока

Статусы «restart», «handler_lost» являются временными. При данных статусах невозможно получить поток, однако переход через эти статусы регистрируется в обычном режиме.

Статус «not_found» является внутренним и будет отправлен для обратной связи, если поток был удален во время обработки. При данном статусе невозможно получить поток.

Статус «deleted» является виртуальным. Поток с таким статусом не может существовать, но этот статус можно увидеть в логах потоков.

4.2.1.1 Таблица переходов от статуса к статусу

Следующая таблица показывает статусы, которые могут быть получены после каждого из перечисленных статусов.

Символ «+» означает, что статус, указанный в первой строке, может стоять после статуса в первом столбце. Пустое поле означает, что нет случаев, когда может возникнуть данный статус.

Символ «-» означает, что в системе нет потока (он не создан или уже удален).

	in_-								handler_-
	-	pending	progress	done	restart	pause	cancel	failure	lost
-		+				+			
pending	+		+		+	+	+		
in_- progress	+	+		+	+	+	+	+	+
done	+	+			+	+			
restart		+				+			
pause	+	+			+		+		
cancel	+	+			+	+			
failure	+	+			+	+			
handler_- lost					+				

* не поддерживается для видеофайлов

4.2.2 Процесс обработки потоков в LUNA Streams

По умолчанию новый поток создается со статусом «pending» и сразу же попадает в очередь обработки. Обработку потока можно отложить, указав статус «pause» при создании.

Как только появляется обработчик свободного потока с запросом на пул из очереди, поток принимается к обработке и ему присваивается статус «in_progress».

После того, как поток был обработан обработчиком FaceStream, ему присваивается статус «done» в случае успеха (актуально для видеофайлов) или «failure», если произошли какие-либо ошибки. Однако статус обработки потока может быть понижен с «in_progress» по следующим причинам:

- отсутствие обратной связи от обработчика потока: сервер понизит рейтинг процесса, а запись со статусом «handler_lost» будет добавлена в логи потока
- замена потока: запись со статусом «restart» будет добавлена в логи потока

Во время процедуры обработки любое изменение статуса потока регистрируется. Таким образом, можно восстановить процесс обработки потоков из логов.

Потоки со статусом «failure» могут быть автоматически перезапущены.

4.2.2.1 Автоматический перезапуск потоков

Возможность автоматического перезапуска потоков актуальна только для потоков со статусом «failure». Параметры автоматического перезапуска (возможность перезапуска, максимальное количество попыток перезапуска, задержка между попытками) задаются пользователем для каждого потока в разделе «autorestart» настроек управления потоками. Параметры и статус автоматического перезапуска можно получить с помощью запроса «get stream».

Ниже перечислены статусы автоматического перезапуска:

- «disabled» - автоматический перезапуск отключен пользователем (отключен параметр «restart»)
- «enabled» - автоматический перезапуск включен, но в данный момент не активен поскольку поток не находится в статуса «failure»
- «in_progress» - автоматический перезапуск в процессе
- «failed» - превышено допустимое количество попыток автоматического перезапуска и ни одна попытка не увенчалась успехом
- «denied» - автоматический перезапуск разрешен пользователем, но не разрешен из-за критической ошибки*, полученной в отчете FaceStream.

* критической ошибкой считается ошибка Failed to authorize in Luna Platform. В будущих версиях список критических ошибок может пополняться.

Процесс выполнения автоматического перезапуска потоков:

- статус потока будет автоматически изменен на «restart», а затем на «pending»
- параметр «current_attempt» (текущая попытка) будет увеличен на 1
- параметр «last_attempt_time» (время последней попытки) будет актуализирован

Возможность автозапуска, максимальное количество попыток перезапуска, задержка между попытками указываются для каждого потока в группе параметров «autorestart».

Поток будет считаться успешно перезапущенным, если по истечении указанного периода времени (задержки) будет иметь статус, отличный от «failure».

Количество одновременных обработок потоков (статусы «pending» и «in_progress») регулируется лицензией, однако в БД LUNA Streams может храниться бесконечное множество потоков, имеющих иной статус, например, «pause».

4.2.3 Группировка потоков

Потоки можно группировать. Группировка призвана объединять потоки с множеством камер в логические группы. Например, можно группировать потоки по территориальному признаку.

Поток может привязан к нескольким группам.

Группа создается с помощью запроса «[create group](#)». Для создания группы нужно указать обязательные параметры «account_id» и «group_name». При необходимости можно указать описание группы.

Поток может быть привязан к группе двумя способами:

- с помощью параметров «group_name» или «group_id» во время создания потока (запрос «[create stream](#)»).
- с помощью запроса «[linker](#)». В запросе необходимо указать идентификаторы потоков и группы, к которой их необходимо привязать.

С помощью запроса «[linker](#)» также можно отвязать потоки от группы.

Если поток был привязан к группе, то при запросе «[get stream](#)» или «[get streams](#)» группа будет отражена в поле «groups».

4.2.4 Описание базы данных LUNA Streams

Общая схема базы данных LUNA Streams приведена ниже.

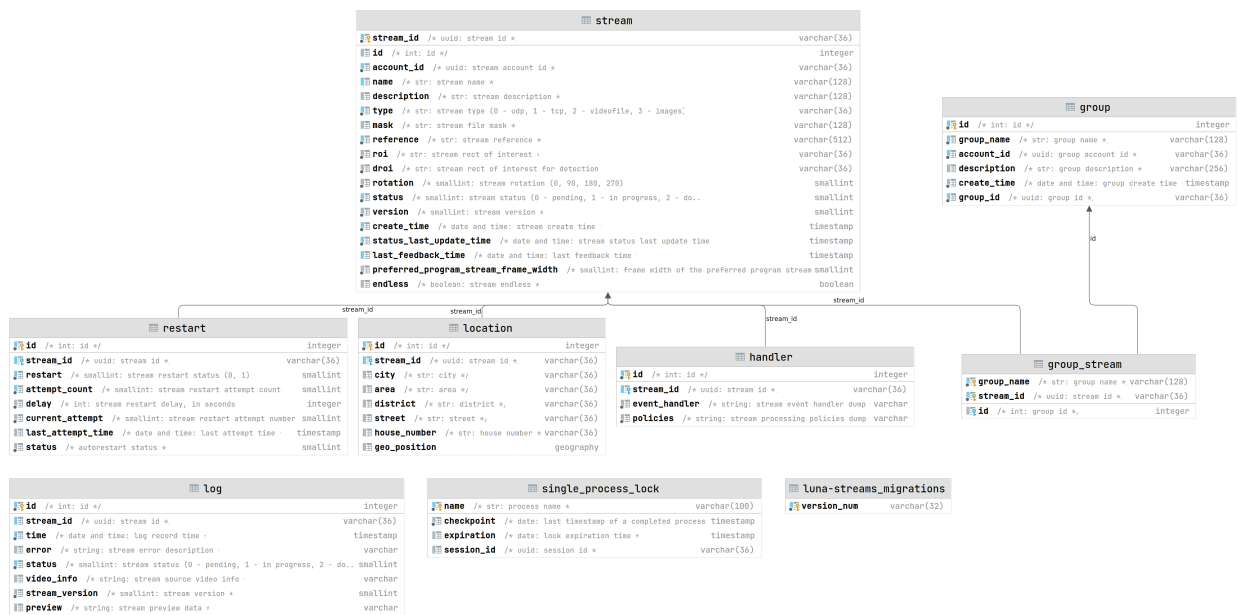


Рис. 4: База данных сервиса LUNA Streams

Описание данных приведено в разделе [«Настройки управления потоками»](#).

4.3 Рекомендации по настройке FaceStream

Данный раздел содержит общие рекомендации по настройке FaceStream.

В разделе указаны названия настроек, в которых настраиваются описываемые параметры.

4.3.1 Рекомендации перед началом настройки

Настройку FaceStream следует выполнять для каждой используемой камеры отдельно. FaceStream должен работать с потоком камеры, находящейся в стандартных рабочих условиях. Данные требования обусловлены следующими причинами:

- Кадры с разных видеокамер могут отличаться по:
 - уровню шума,
 - размеру кадра,
 - освещённости,
 - размытости и т. п.
- Настройки FaceStream зависят от условий освещения, поэтому будут отличаться для камер, расположенных в тёмном и светлом помещении;
- Производительность FaceStream зависит от количества лиц или тел, одновременно находящихся в кадре. Поэтому настройки для камеры, которая детектирует одно лицо в 10 секунд, будут отличаться от настроек для камеры, которая детектирует 10 лиц в секунду;
- Количество детекций лиц или тел и качество этих детекций зависят от правильного расположения камеры. Если камера расположена под неправильным углом, то объекты не будут детектироваться на кадрах или не могут быть использованы для дальнейшей обработки, например, из-за слишком больших углов поворота головы на кадрах;
- Лица или тела в зоне видимости камеры могут быть частично или полностью перекрыты посторонними предметами или на фоне могут располагаться объекты, мешающие алгоритмам распознавания.

Камера может быть расположена так, что освещение или условия съёмок изменяются в течение дня. Рекомендуется протестировать работу FaceStream при разных условиях и выбрать оптимальный режим, обеспечивающий надёжную работу FaceStream при любых условиях.

Скорость обработки кадров для видео можно задать с помощью параметра `«real_time_mode_fps»`.

Список протестированных с FaceStream камер можно найти в разделе [«Приложение А. Совместимость с камерами»](#).

4.3.2 Настройка производительности FaceStream

Описанные ниже настройки оказывают наибольшее влияние на производительность FaceStream.

4.3.2.1 Уменьшение области поиска лиц

Не все области кадра содержат лица. Кроме того, не все лица в кадре имеют необходимый размер и качество. Например, размер лиц на заднем фоне может быть меньше допустимого, а лица по краям кадра могут иметь слишком большие углы поворота.

Параметр «roi» (настройки управления потоками, группа параметров «data»), позволяет задать прямоугольную область, в которой будет выполняться поиск лиц.

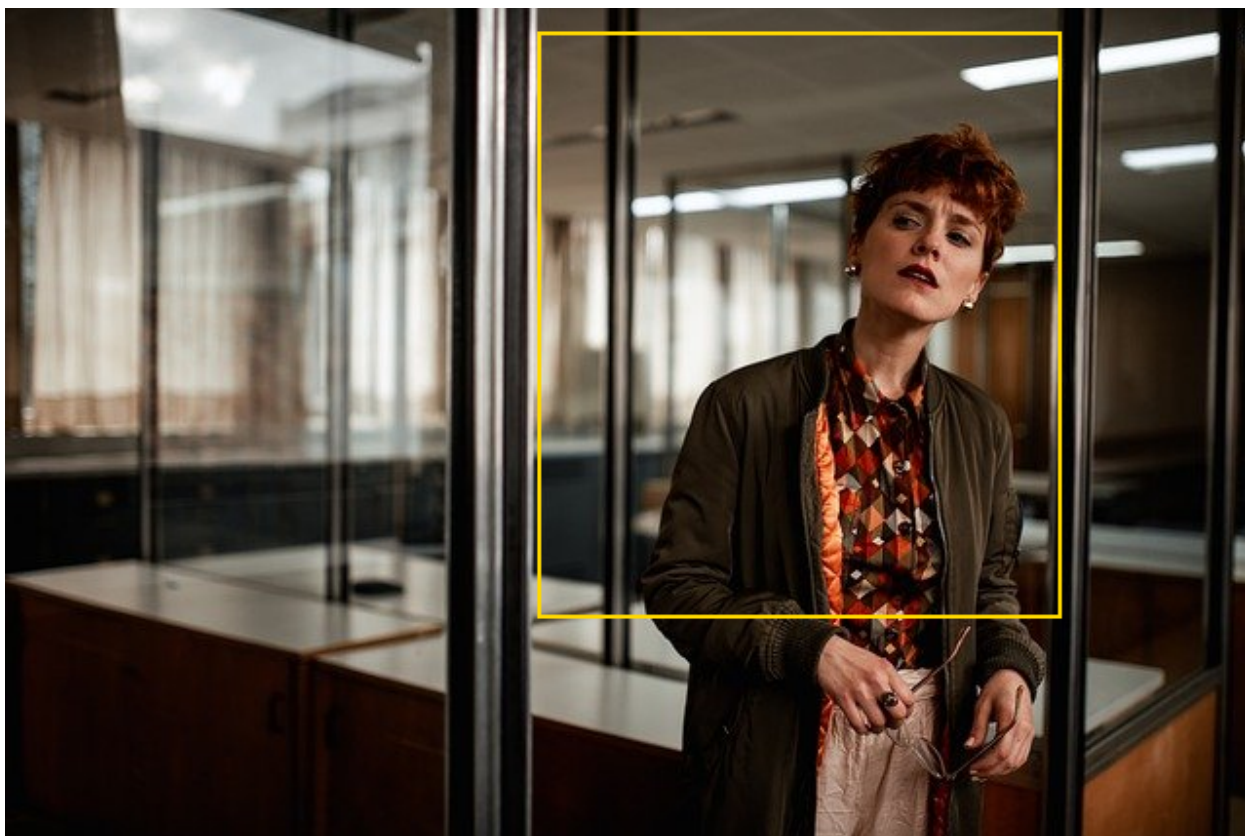


Рис. 5: Исходный кадр с зоной ROI

Указанная прямоугольная область вырезается из кадра и именно это изображение в дальнейшем обрабатывает FaceStream.



Рис. 6: Вырезанная область кадра, используемая для обработки в FaceStream

Чем меньше зона поиска, тем меньше ресурсов потребляется на обработку каждого кадра.

Правильное использование параметра «roi» существенно повышает производительность работы FaceStream.

Параметр необходимо использовать только при работе с лицами.

4.3.2.2 Масштабирование кадра

Масштабирование кадра перед началом обработки позволяет значительно увеличить производительность FaceStream. Масштабирование кадра включается с помощью параметра [«detector-scaling»](#).

Подходящий для работы размер кадра следует подобрать с помощью параметра [«scale-result-size»](#). Параметр [«scale-result-size»](#) задаёт максимальный размер кадра после масштабирования по наибольшей из сторон кадра. Если исходный кадр имел размер 1920 на 1080 и значение [«scale-result-size»](#) равно 640, то FaceStream будет обрабатывать кадр размером 640 на 360. Для подбора оптимального значения [«scale-result-size»](#) следует постепенно уменьшать масштаб кадра и проверять, происходит ли на кадре детектирование лиц или тел. Следует выставить минимальный размер изображения, при котором все объекты в интересующей области детектируются.

См. пример использования параметра [«scale-result-size»](#) в соответствующем разделе.

4.3.3 Определение области с движением

За определение области с движением отвечают три параметра, задаваемые в [настройках TrackEngine](#):

- [frg-subtractor](#) - включение режим учёта перемещений в кадре;
- [frg-regions-alignment](#) - задание выравнивания для областей с движением;
- [frg-regions-square-alignment](#) - установка равных ширины и высоты области с движением.

Ниже приведены рекомендуемые значения для данных параметров при использовании CPU и GPU:

	frg-subtractor	frg-regions-alignment	frg-regions-square-alignment
Рекомендованное значение при использовании CPU	1	0	0
Рекомендованное значение при использовании GPU	1	360	0

4.3.4 Обработка пакетов кадров

За пакетную обработку кадров отвечают три параметра, задаваемые в [настройках TrackEngine](#):

- [batched-processing](#) - включение пакетной обработки кадров;
- [min-frames-batch-size](#) - минимальное количество кадров, которое следует накопить со всех камер перед обработкой;
- [max-frames-batch-gather-timeout](#) - время между обработкой пакетов кадров.

4.3.4.1 Минимальный размер лица

В настройках FaceEngine следует настроить параметр «minFaceSize», задающий минимальный размер лица для выполнения детектирования.

Следует задавать наибольший возможный минимальный размер лица, т. к. чем больше лицо, тем меньше ресурсов требуется на выполнение детекции.

Важно, что размер лица будет зависеть от фактического размера кадра, заданного параметром «[scale-result-size](#)» (настройки TrackEngine). Лицо, которое имело размер 100 пикселей на кадре 1280x960, будет иметь размер 50 пикселей на кадре размером 640x480.

4.3.5 Общая информация о настройке

4.3.5.1 Работа с треком

Для каждого найденного лица или тела создаётся новый трек. В рамках этого трека определяются и отправляются лучшие кадры.

В общем случае трек прерывается, когда объект больше не может быть найден на кадре.

Если трек был прерван, а человек вновь появился в кадре, то для него будет создан новый трек.

Возможна ситуация, когда два лица или тела пересекаются в кадре (один человек проходит за спиной другого). В этом случае треки для обоих людей обрываются и создаются новые треки.

Возможна ситуация, когда человек отворачивается или его лицо или тело временно перекрывается. В этом случае трек можно не обрывать сразу, а указать параметр `«skip-frames»` в настройках TrackEngine. Он задаёт количество кадров, в течение которых система будет ожидать повторного появления объекта в области, где он исчез.

При работе с треком также полезно использовать параметр `«detector-step»`, позволяющий указать количество кадров, на которых будет выполняться редетекция лица или тела в заданной области до выполнения детекции лица или тела.

4.3.5.2 Отправка лучшего кадра

Группа параметров `«sending»` (настройки управления потоками) позволяет задать параметры для отправки лучшего кадра. FaceStream отправляет полученные лучшие снимки в LUNA PLATFORM (см. `«Перечень приоритетных настроек»`).

Для увеличения точности распознавания можно отправлять для одного лица или тела не один, а сразу несколько лучших снимков. Для этого нужно включить параметр `«number_of_bestshots_to_send»` (настройки управления потоками).

LUNA PLATFORM позволяет агрегировать данные лучшие снимки и создавать на их основе биометрический шаблон более высокого качества.

Если за указанный период или до окончания трека не набралось необходимое количество лучших снимков, то отправляются только те лучшие снимки, которые были собраны.

Значения параметров `«time_period_of_searching»` (настройки управления потоками) и `«silent_period»` (настройки управления потоками) могут быть заданы в секундах или кадрах. Для выбора единиц измерения используется параметр `«type»`.

Далее перечислены основные варианты настроек параметров `«time_period_of_searching»` и `«silent_period»` группы `«sending»` из настроек управления потоками.

- Лучший снимок отправляется только после того, как трек закончился и человек вышел из зоны видимости камеры.

Таким образом анализируются все кадры с лицом или телом человека и выбирается лучший кадр.

```
time_period_of_searching = -1  
silent_period = 0
```

- Требуется получить лучший кадр как можно быстрее и отправлять лучшие снимки с указанной периодичностью.

Например, если злоумышленник зашёл в супермаркет, требуется сразу отправить лучший кадр для сравнения с чёрным списком.

Данный режим так же используется для демонстрации работы FaceStream в реальном времени.

В конце трека лучший кадр будет отправлен, даже если заданный период не закончился.

```
time_period_of_searching = 3  
silent_period = 0
```

- Требуется быстро отправить лучший кадр, а потом отправить заново, только если человек долго находится в кадре.

```
time_period_of_searching = 3  
silent_period = 20
```

- Требуется быстро отправить лучший кадр и больше никогда не отправлять с этого трека лучший кадр.

```
time_period_of_searching = 3  
silent_period = -1
```

4.3.5.3 Фильтрация кадров

Фильтрация кадров лиц выполняется по трём основным критериям (задаются в настройках для управления потоками”):

- Углом наклона головы («[detection_yaw_threshold](#)», «[detection_pitch_threshold](#)», «[detection_roll_threshold](#)»).

Для угла поворота (yaw) задаются два дополнительных параметра «[yaw_number](#)» и «[yaw_collection_mode](#)», которые позволяют уменьшить вероятность возникновения ошибки, когда вместо большого угла наклона возвращается угол «0»;

- Качеству кадра для его дальнейшей обработки («min_score»);
- Перекрытию области рта ("mouth_occlusion_threshold").

Если кадр не прошёл фильтрацию хотя бы по одному из указанных фильтров, то он не может быть выбран в качестве лучшего кадра.

Если задан параметр «number_of_bestshots_to_send», то кадр, прошедший фильтрацию, добавляется в массив лучших снимков для отправки. Если уже набрано необходимое количество лучших снимков для отправки, то один из этих снимков, имеющий худшее качество кадра (score), будет заменён новым лучшим кадром, если его качество выше.

Фильтрация кадров тел выполняется только по одному критерию - «min_score».

4.3.5.4 Работа со СКУД

Работа со СКУД ведется только с лицами.

При работе со СКУД используются настройки «primary_track_policy» из настроек управления потоками. Они позволяют включить режим работы только с одним лицом, размер которого является самым большим. Предполагается, что интересующее лицо находится близко к камере.

Трек самого большого лица в кадре становится главным треком. Все остальные лица в кадре детектируются, но не обрабатываются. Лучшие снимки для них не отправляются.

Как только другое лицо достигнет большего размера, чем лицо главного трека, трек этого лица станет новым главным треком и обработка будет выполняться для него.

Включение режима выполняется настройкой «use_primary_track_policy».

Определение лучшего кадра начинается только когда лицо достигает размера (по вертикали), заданного в параметре «best_shot_min_size». Кадры с лицами меньшего размера не могут стать лучшими снимками.

Когда размер детекции лица по вертикали достигает значения установленного в «best_shot_proper_size», сразу же выполняется отправка лучшего кадра.

Значения «best_shot_min_size» и «best_shot_proper_size» задаются в зависимости от используемой видеокамеры и её расположения.

Далее приведены примеры настройки параметров группы «sending» из настроек управления потоками для работы со СКУД.

- Турникет откроется только один раз. Для повторного открытия турникета потребуются обогреть трек (отойти из зоны видимости видеокамер).

```
time_period_of_searching = -1
silent_period = 0
```

- Турникет будет открываться с некоторой периодичностью (в данном случае каждые три секунды), если человек встал прямо перед ним.

```
time_period_of_searching = 3  
silent_period = 0
```

При включённом параметре «use_primary_track_policy» лучший кадр никогда не отправляется при окончании трека.

4.4 Форматы, стандарты сжатия видео и протоколы

FaceStream использует библиотеку FFmpeg для конвертации видео и получения потока с использованием различных протоколов. Далее перечислены все основные форматы, стандарты сжатия видео и протоколы, которые были протестированы при работе с FaceStream.

FFmpeg поддерживает большее количество форматов и стандартов сжатия видео. Они не перечислены в данном разделе, т. к. крайне редко используются при работе с FaceStream.

4.4.1 Форматы видео

Основные форматы видео, обрабатываемые FaceStream:

- AVI,
- MP4,
- MOV,
- MKV,
- FLV.

4.4.2 Стандарты сжатия видео

Основные стандарты сжатия видео, с которыми работает FaceStream:

- MPEG4,
- MS MPEG4,
- MS MPEG4v2,
- MJPEG,
- H.264,
- H.265.

4.4.3 Протоколы

Основные протоколы, используемые FaceStream для получения данных:

- HTTP,
- RTP,
- RTSP,
- RTMP,
- TCP,
- HLS,
- UDP.

4.5 Потребление памяти при работе FaceStream

В данном разделе перечислены причины увеличения потребления оперативной памяти при работе FaceStream.

1. Каждый подключенный поток даёт прирост потребления памяти. Количество потребляемой памяти зависит от настроек, заданных для FaceStream:
 - числа Ffmpeg потоков в параметре «[ffmpeg_threads_number](#)» (настройки управления потоками),
 - размера кэша изображений в параметре «[stream_images_buffer_max_size](#)» (настройки FaceStream),
 - размеров буферов в параметре «frames-buffer-size» (настройки TrackEngine).
2. Если количество потоков, заданных в параметре «[ffmpeg_threads_number](#)», больше «1», то значительно увеличивается потребляемая память. При этом прирост потребления происходит крайне медленно и замечен лишь через несколько часов работы.

Для RTSP потоков можно выставить значение параметра «[ffmpeg_threads_number](#)» равным «0» или «1». В этом случае рост памяти не выявлен.
3. Потребление памяти растёт после запуска FaceStream. Рост происходит в течение 1-2 часов. Это связано с наполнением кэшей (см. пункт 1). Если новые потоки не создаются и не выполняется пункт 2, то потребление памяти перестаёт расти.
4. Потребление памяти растёт при включённых настройках в секции Debug (настройки FaceStream и TrackEngine).

4.6 Мониторинг

Мониторинг реализован как отправка данных в [«InfluxDB OSS 2»](#). Мониторинг включён в сервисах LUNA PLATFORM по умолчанию, но может быть отключен.

Мониторинг осуществляется только для сервисов LUNA PLATFORM. Для FaceStream мониторинг не используется.

Мониторинг возможен для двух типов событий: запрос (все запросы) и ошибка (только неудавшиеся запросы).

Каждое событие - это точка временного ряда. Точка представлена с использованием следующих данных:

- тип события (запросы или ошибки)
- отметка времени начала запроса
- теги
- поля

Тег - это индексированные данные в хранилище. Он представлен в виде словаря, где

- keys - имена тегов (строка),
- values - строка, целое число или число с плавающей запятой.

Поле представляет собой неиндексированные данные в хранилище. Оно представлено в виде словаря, где

- keys - имена полей (строка),
- values - строка, целое число или число с плавающей запятой.

См. более подробную информацию в руководстве администратора LUNA PLATFORM.

4.6.1 InfluxDB OSS 2

Для использования InfluxDB OSS 2 необходимо:

- Установить БД. См. «Запуск контейнера InfluxDB OSS 2» в руководстве по установке.
- Зарегистрироваться в БД. InfluxDB имеет пользовательский интерфейс, в котором можно зарегистрироваться. Посетите `<server_ip>:<influx_port>`.
- Настроить отображение информации мониторинга в пользовательском интерфейсе. Описание этой настройки не приведено в документации.

4.6.1.1 Настройка InfluxDB

Ниже приведены настройки InfluxDB:

Таблица 4: Настройки InfluxDB

Название настройки	Тип	Описание
send_data_for_monitoring	integer	Активирует мониторинг для сервиса.
use_ssl	integer	Позволяет использовать HTTPS протокол для соединения с InfluxDB (0 – не использовать, 1 – использовать).
flushing_period	integer	Частота отправки данных мониторинга в InfluxDB.
host	integer	IP адрес InfluxDB.
port	integer	Порт InfluxDB.
organization	String	Название организации, указанное при регистрации.
token	String	Токен, полученный после регистрации.
bucket	String	Название бакета.

4.7 Интерфейс воспроизведения потока

В FaceStream доступна возможность просмотра потока в режиме реального времени. Для просмотра потока необходимо ввести следующий адрес в браузерную строку после начала обработки потока приложением FaceStream:

```
http://127.0.0.1:34569/api/1/streams/preview/<stream_id>.
```

При появлении объектов в поле видимости камеры, FaceStream отображает их определенным образом.

Ограничивающий прямоугольник **желтого цвета** возникает если для детекции не пройден хотя бы один порог из «[detection_yaw_threshold](#)», «[detection_pitch_threshold](#)» или «[detection_roll_threshold](#)».

Ограничивающий прямоугольник **красного цвета** возникает если оценка приемлемости детекции ниже, чем значение, заданное в параметре «[min_score](#)».

Ограничивающий прямоугольник **синего цвета** возникает при детекции (редетекции) или трекинге объекта.

Ограничивающий прямоугольник **зеленого цвета** возникает во всех остальных случаях при соблюдении всех условий.

5 Перечень приоритетных настроек

Для отправки фотоизображений в LUNA PLATFORM в первую очередь необходимо настроить FaceStream на работу с лицами или телами (см. настройки для переключения режима обнаружения ниже), а также настроить основные параметры, необходимые для корректной работы приложения. Все настройки разделены следующим образом:

- Настройки FaceStream - задаются в секции «FACE_STREAM_CONFIG» в Configurator или в конфигурационном файле «fs3Config.conf».
- Настройки управления потоками - задаются в запросе с телом в формате JSON к ресурсу [«/streams»](#) в сервис LUNA Streams.
- Настройки TrackEngine - задаются в секции «TRACK_ENGINE_CONFIG» в Configurator или в конфигурационном файле «trackengine.conf».

См. подробное описание нижеперечисленных параметров в соответствующих разделах.

Доступны следующие общие параметры для отправки и лиц и тел:

Таблица 5: Настройки FaceStream

Параметр	Описание
sending > async_requests	Позволяет переключаться между режимами асинхронной и синхронной отправки запросов в LUNA PLATFORM
sending > send_source_frame	Позволяет включить отправку исходного кадра в LUNA PLATFORM
sending > jpeg_quality_level	Позволяет задать степень сжатия исходного кадра

Таблица 6: Настройки управления потоками

Параметр	Описание
event_handler > frame_store	Позволяет задать URL сервиса Image Store для отправки исходного кадра

5.1 Настройки для работы с лицами

Основные параметры, которые должны быть указаны для отправки фотоизображений лиц в соответствующий сервис, указаны ниже.

Таблица 7: Настройки TrackEngine

Параметр	Описание
<code>use-face-detector</code>	Позволяет обнаруживать лица - «1»
<code>use-body-detector</code>	Позволяет обнаруживать тела - «0»

Таблица 8: Настройки FaceStream

Параметр	Описание
<code>lunastreams > api_version</code>	Позволяет задать версию API сервиса LUNA Streams для отправки запросов в LP - «1»
<code>lunastreams > origin</code>	Полный сетевой путь до сервиса LUNA Streams
<code>sending > request_type</code>	Тип запроса для отправки изображений в LP - «jpeg»
<code>sending > portrait_type</code>	Формат передачи изображения - «warp»
<code>sending > aggregate_attr_requests</code>	Позволяет включать агрегацию лучших кадров для получения одного биометрического шаблона в LUNA PLATFORM - «true» или «false»

Таблица 9: Настройки управления потоками

Параметр	Описание
<code>data > type</code>	Тип источника сигнала (tcp, udp, videofile, images) - «строка»
<code>account_id</code>	«Luna_account_id», к которому относится выполняемый запрос - «строка в формате UUID»

Параметр	Описание
<code>event_handler > origin</code>	полный сетевой путь до LP 5 - «http://<luna_api_adress>:5000/». В примере указан порт «5000» для сервиса API, который используется по умолчанию
<code>event_handler > api_version</code>	версия API LP 5. Поддерживается версия API - 6
<code>event_handler > bestshot_handler > handler_id</code>	обработчик LP, позволяющий гибко настроить обработку лиц - «строка в формате UUID»

Подробную информацию об обработчиках можно найти в документации [APIReferenceManual.html](#) в комплекте поставки LUNA PLATFORM 5.

5.2 Настройки для работы с телами

Основные параметры, которые должны быть указаны для отправки фотоизображений тел в соответствующий сервис, указаны ниже.

Таблица 10: Настройки TrackEngine

Параметр	Описание
<code>use-face-detector</code>	Позволяет обнаруживать лица - «1»
<code>use-body-detector</code>	Позволяет обнаруживать тела - «0»

Таблица 11: Настройки FaceStream

Параметр	Описание
<code>lunastreams > api_version</code>	Позволяет задать версию API сервиса LUNA Streams для отправки запросов в LP - «1»
<code>lunastreams > origin</code>	Полный сетевой путь до сервиса LUNA Streams
<code>sending > aggregate_attr_requests</code>	Позволяет включать агрегацию лучших кадров для получения одного биометрического шаблона в LUNA PLATFORM - «true» или «false»

Параметр	Описание
<code>sending > minimal_body_track_length_to_send</code>	Данный параметр включает отправку детекций с координатами тела человека - x, y, width и height и устанавливает количество детекций, меньше значения которого они не будут отправляться - «3»
<code>sending > detection_path_length</code>	Устанавливает максимальное количество детекций для параметра «minimal_body_track_length_to_send» - «100»

Таблица 12: Настройки управления потоками

Параметр	Описание
<code>data > type</code>	Тип источника сигнала (tcp, udp, videofile, images) - «строка»
<code>account_id</code>	«Luna_account_id», к которому относится выполняемый запрос - «строка в формате UUID»
<code>event_handler > origin</code>	полный сетевой путь до LP 5 - «http://<luna_api_adress>:5000/». В примере указан порт «5000» для сервиса API, который используется по умолчанию
<code>event_handler > api_version</code>	версия API LP 5. Поддерживается версия API - 6
<code>event_handler > bestshot_handler > handler_id</code>	обработчик LP, позволяющий гибко настроить обработку тел - «строка в формате UUID»
<code>event_handler > detection_handler > handler_id</code>	динамический обработчик LP, позволяющий прикрепить координаты тела к событию - «строка в формате UUID»

Подробную информацию об обработчиках можно найти в документации [APIReferenceManual.html](#) в комплекте поставки LUNA PLATFORM 5.

6 Настройки управления потоками

Приложение поддерживает одновременную работу с несколькими источниками потоков.

Параметры для управления потоками задаются сервисе [LUNA Streams](#). Сервис позволяет создавать и хранить потоки в базе данных LUNA Streams.

Поддерживаются несколько типов источников:

- tcp, udp – источники видеосигнала реального времени. Это могут быть как usb-камеры, так и IP-камеры (посредством протокола RTSP);
- videofile – видеофайлы;
- images – набор кадров в виде отдельных файлов изображений.

Обратите внимание, что настройки управления потоками не хранятся в сервисе LUNA Configurator и могут быть заданы только с помощью HTTP запросов к сервису LUNA Streams. Подробное описание запросов и примеры запросов можно найти в [спецификации Open API LUNA Streams](#). Настройки LUNA Streams, задаваемые в LUNA Configurator, описаны в разделе «Настройки LUNA Streams».

6.1 Параметр «account_id»

Параметр задаёт значение обязательного поля «account_id», которое передаётся в заголовке запроса в LUNA PLATFORM 5 в сервис API.

Account ID задаётся в формате UUID4. Требования в Account ID можно найти в документации LUNA PLATFORM 5.

Параметр используется для привязки получаемых данных к конкретному пользователю.

```
"account_id" : {  
  "value" : "aaba1111-2111-4111-a7a7-5caf86621b5a",  
  "description" : "Luna_account_id header value for Luna api version 6 or  
    higher (' ' by default)."  
}
```

6.2 Параметр «name»

Имя потока. Служит для идентификации источника отправляемых кадров.

```
"name": "stream_0",
```

Записывается в поле «source» генерируемого события LP.

6.3 Параметр «description»

Пользовательское описание потока.

```
"description": "Stream on Arbat street",
```

Данный параметр не записывается в генерируемое событие LP.

6.4 Группа параметров «data»

Ниже перечислены основные параметры, необходимые для настройки потока.

6.4.1 Параметр «type»

Тип передачи потока. Приложение может использовать один из следующих типов передачи потока:

- **tcp** - сетевой протокол для приема видеоданных
- **udp** - сетевой протокол для приема видеоданных
- **images** - набор кадров в виде отдельных файлов изображений
- **videofile** - видеофайл

```
"type": "tcp",
```

Протокол TCP реализует механизм контроля ошибок, позволяющий минимизировать потерю информации и пропуски опорных кадров ценой увеличения сетевой задержки. **Опорные кадры** являются основой различных алгоритмов сжатия, используемых в видеокодеках (например, h264). Только опорные кадры содержат достаточное количество информации для полного восстановления (декодирования) изображения, в то время как **промежуточные кадры** содержат лишь отличия между соседними опорными кадрами.

В условиях вещания по сети существует риск потери пакетов из-за несовершенства каналов связи. В случае потери пакета, содержащего данные опорного кадра, невозможно корректно декодировать фрагмент потока. Как следствие, возникают характерные артефакты, легко различимые визуально. Эти артефакты не позволяют детектору лиц работать в штатном режиме.

Протокол UDP не реализует механизма контроля ошибок, поэтому поток не защищен от повреждения. Использование данного протокола рекомендуется только при наличии высококачественной сетевой инфраструктуры.

При большом количестве потоков (10 и более) настоятельно рекомендуется использовать **протокол UDP**. При использовании **протокола TCP** могут возникнуть проблемы с чтением потоков.

6.4.2 Параметр «reference»

Полный путь к источнику (для типа «tcp»/«udp»):

```
"reference": "rtsp://some_stream_address"
```

Номер USB устройства (для типа «tcp»/«udp»):

```
"reference": "/dev/video0"
```

Для использования USB устройства, необходимо указать флаг `--device` с адресом USB устройства при запуске Docker-контейнера FaceStream. См. раздел «Ключи запуска» в руководстве по установке FaceStream.

Полный путь к видеофайлу (для типа «videofile»):

```
"reference": https://127.0.0.1:0000/super_server/
```

Полный путь к директории с изображениями (для типа «images»):

```
"reference": "/example1/path/to/images/"
```

Для использования видеофайлов и изображений необходимо их предварительно перенести в Docker-контейнер.

6.4.3 Параметр «roi»

Данный параметр используется только для работы с лицами.

ROI задаёт область интереса в которой происходит детекция и сопровождение лица на кадре.

Указанная прямоугольная область вырезается из кадра и именно это изображение в дальнейшем обрабатывает FaceStream.

Правильное использование параметра «roi» существенно повышает производительность работы FaceStream.

Область интереса ROI на исходном кадре, задается в пикселях как массив вида [x, y, ширина, высота], где (x, y) – координаты верхней левой точки области интереса ROI. Система координат на изображении задается аналогично рисунку ниже.

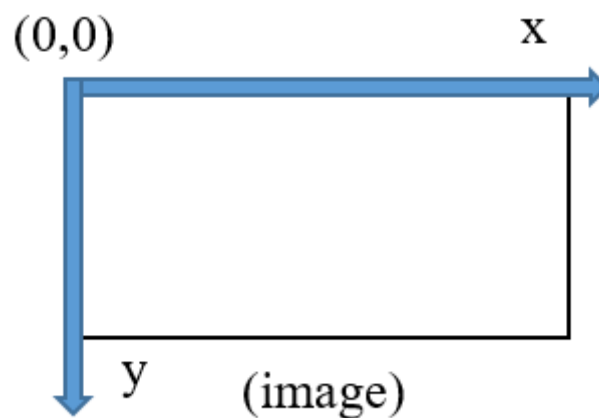


Рис. 7: ROI

При значениях ширины и высоты, равных «0», областью интереса считается весь кадр.

```
"roi": [0, 0, 0, 0],
```

6.4.4 Параметр «droi»

Данный параметр используется только для работы с лицами.

Параметр задаёт область интереса внутри области ROI. Детектирование лиц выполняется в области ROI, но лучший кадр выбирается только в области DROI. Детекция лица должна полностью находиться внутри области DROI, чтобы кадр рассматривался в качестве лучшего.

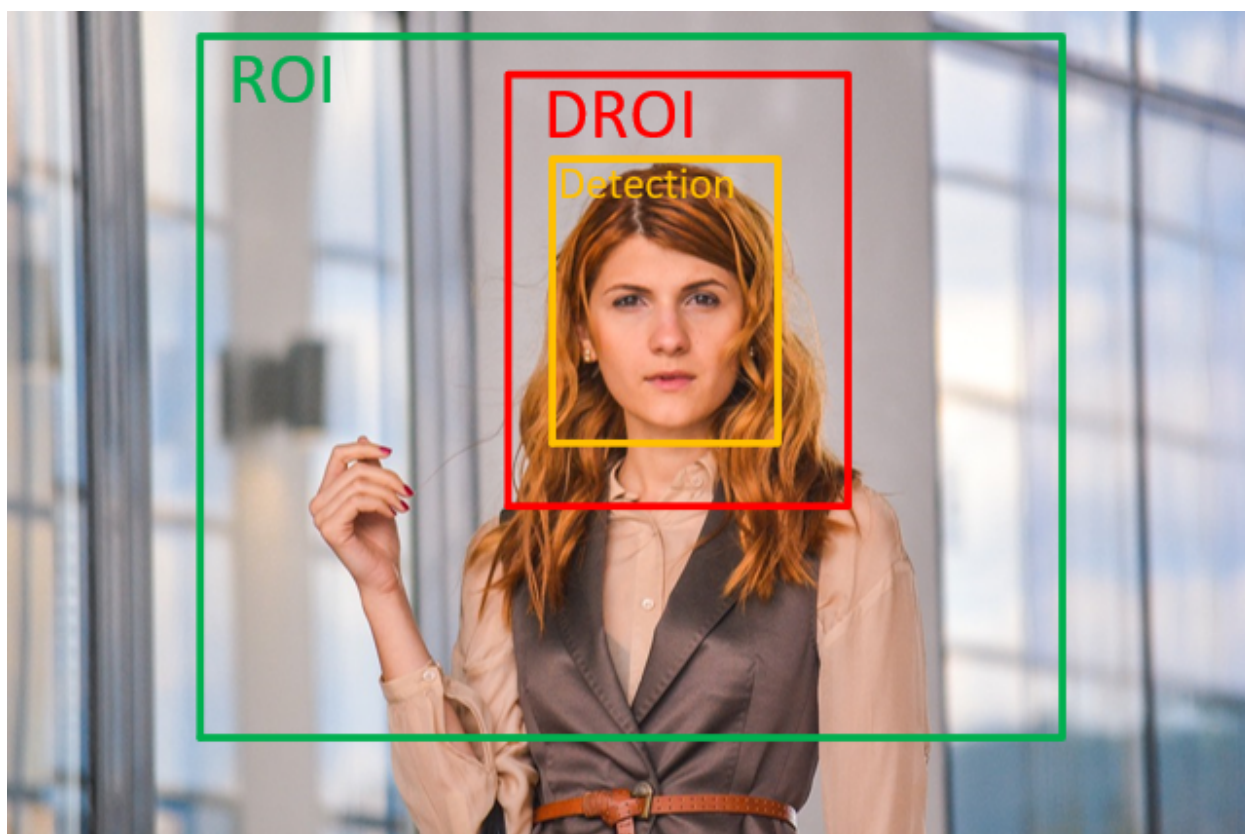


Рис. 8: DROI

DROI рекомендуется использовать при работе со СКУД и при включённом режиме «use_mask_liveness_filtration».

Например, если рядом стоит несколько турникетов и каждый из них должен искать лицо только в небольшой зоне и одновременно осуществлять проверку Liveness. Использование DROI позволяет ограничить область для определения лучшего кадра и при этом не терять информацию о заднем плане.

Область интереса DROI на исходном кадре задается в пикселях как массив вида [x,y, ширина, высота], где (x, y) – координаты верхней левой точки области интереса DROI. При расчете DROI нужно учитывать то, что данная область интереса рассчитывается относительно исходного кадра, а не относительно ROI.

Система координат задается аналогично рисунку ниже.

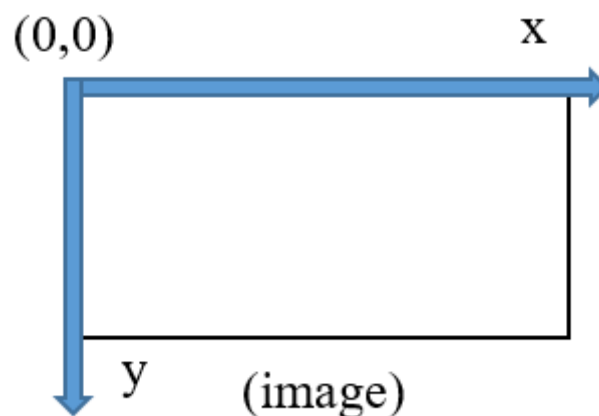


Рис. 9: Система координат DROI

Если размер ROI изменяется, а размер DROI остаётся в значении по умолчанию (0, 0, 0, 0), то DROI не учитывается. Если изменить размер DROI, то он будет учитываться при выборе лучшего кадра.

```
"droi": [0, 0, 0, 0],
```

6.4.5 Параметр «rotation»

Угол поворота изображения с источника. Используется в случае, если входящий поток повернут, например, если камера установлена на потолке. Поворот выполняется по часовой стрелке.

```
"rotation": 0,
```

6.4.6 Параметр «preferred_program_stream_frame_width»

Данный параметр используется только для типов **tcp** или **udp** и предназначен для работы с протоколами, в которых подразумевается наличие нескольких каналов с разными битрейтами и разрешением (например, HLS).

Если у потока существует несколько таких каналов, то данный параметр позволит выбрать из всех каналов цельного потока тот канал, ширина кадра которого ближе к значению, указанному в данном параметре.

Например, имеется 4 канала, ширины кадров которых равны 100, 500, 1000 и 1400. Если параметр «preferred_program_stream_frame_width» равен «800», то будет выбран канал, ширина кадра которого равняется 1000.

Если у потока всего один канал, то данный параметр будет игнорироваться.

Значение по умолчанию равно 800.

```
"preferred_program_stream_frame_width": 800
```

6.4.7 Параметр «endless»

Данный параметр позволяет управлять перезагрузкой потока при получении ошибки сети (ошибка определяется системой как маркер конца файла eof).

Параметр доступен только для типов источника сигнала «udp» и «tcp».

Если параметр `endless` принимает значение `true`, то в случае получения eof и успешного переподключения, обработка потока будет продолжена. Если все попытки переподключения не удалось (см. группу параметров «[healthcheck](#)»), то поток примет статус «failure». Если же параметр принимает значение `false`, то обработка потока не будет продолжена и поток примет статус «done».

При трансляции видеофайла предполагается использование значения `false`. Это позволит избежать повторной обработки уже обработанного фрагмента видеофайла при получении eof. Если же при трансляции видеофайла значение параметра `endless` будет `true`, то после окончания обработки видеофайл начнет обрабатываться с начала.

```
"endless": true
```

6.4.8 Параметр «mask»

Параметр `Mask` используется только для типа **images** и является обязательным параметром.

Маска имён файлов в директории с изображениями. Маска позволяет `FaceStream` понять, какие файлы из указанной директории следует использовать и в каком порядке.

Если задать маску «`Img_%02d.jpg`», то `FaceStream` будет обрабатывать файлы, имена которых состоят из: Префикс (`Img_`) + двузначное число (`%02d`) + формат (`.jpg`)

Изображения, подходящие под маску, будут обрабатываться по очереди изображения:

- `Img_00.jpg`
- `Img_01.jpg`
- `Img_02.jpg`
- `Img_03.jpg`

Другой пример маски – `Photo-%09d.jpg`. Подходящие под маску файлы:

- `Photo-000000000.jpg`

- Photo-0000000001.jpg
- Photo-0000000002.jpg
- Photo-0000000003.jpg

FaceStream обрабатывает файлы по порядку нумерации и не пропускает несуществующие файлы. Если следующий по нумерации файл отсутствует, FaceStream прекращает обработку.

Указанная в примере маска «example1_%04d.jpg» приведёт к обработке изображений, название которых состоит из префикса example1_ и порядкового номера кадра размером в 4 символа (например: example1_0001.jpg, example1_0002.jpg и т. д.).

```
"mask": "example1_%04d.jpg"
```


6.5 Группа параметров «event_handler»

Данная группа параметров определяет параметры обработчика, создаваемого в LUNA PLATFORM, с помощью которого будут обрабатываться потоки. Для лиц и тел следует использовать различные обработчики. Обработчик должен быть создан в LP 5 заранее.

Подробную информацию об обработчиках см. в руководстве администратора LUNA PLATFORM.

6.5.1 Параметр «origin»

Полный сетевой путь к развернутой LUNA PLATFORM 5, включающую сервисы LUNA Handlers и LUNA Events, необходимыми для генерации события по обработчику.

```
"origin": "http://<luna_address>:<port>/"
```

<luna_address> - адрес сервиса LUNA API,

<port> - порт, используемый сервисом API. По умолчанию - 5000.

6.5.2 Параметр «api_version»

Версия API для генерации события в LUNA PLATFORM. На настоящий момент поддерживается 6 версия API.

6.5.3 Параметр «handler_id» группы «bestshot_handler»

Параметр позволяет использовать внешний статический «handler_id» LUNA PLATFORM для обработки биометрических образцов лиц или тел по заданным правилам. При использовании данного обработчика, LUNA PLATFORM генерирует событие, которое содержит всю информацию, полученную из FaceStream и обрабатывает её в соответствии с правилами обработки.

```
"handler_id": "aaba1111-2111-4111-a7a7-5caf86621b5a",
```

Handler должен быть создан в LP 5 заранее.

6.5.4 Параметр «handler_id» группы «detection_handler»

Параметр позволяет использовать внешний динамический «handler_id» LUNA PLATFORM для работы с **телами**. Данный параметр используется совместно с параметром [minimal_body_track_length_to_send](#) и их взаимодействие позволяет отправлять детекции с координатами тела человека (x, y, width и height) вместе с лучшими снимками.

```
"handler_id": "aaba1111-2111-4111-a7a7-5caf86621b5a",
```

Данный параметр не является обязательным и требуется только при необходимости детекции координат тела человека.

Подробную информацию о динамических обработчиках см. в руководстве администратора LUNA PLATFORM.

6.5.5 Параметр «frame_store»

В данном параметре задаётся URL для сохранения исходных кадров лиц или тел в LUNA PLATFORM 5.

В качестве URL можно указать либо адрес до бакета сервиса LUNA Image Store, либо адрес к ресурсу «/images» сервиса LUNA API. При указании адреса к ресурсу «/images», исходный кадр будет сохранен под идентификатором «image_id».

Для отправки кадра следует включить опцию [send_source_frame](#).

Пример адреса до бакета сервиса LUNA Image Store:

```
"frame_store": "http://127.0.0.1:5020/1/buckets/<frames>/images"
```

127.0.0.1 - IP адрес, где развернут сервис LUNA Image Store;

5020 - порт сервиса LUNA Image Store по умолчанию;

1 - версия API сервиса LUNA Image Store;

<frames> - имя бакета сервиса LUNA Image Store, в котором нужно сохранить изображение лица или тела. Бакет должен быть создан пользователем предварительно.

Пример команды создания бакета с именем «source-images»:

```
curl -X POST http://127.0.0.1:5020/1/buckets?bucket=source-images
```

Пример адреса к ресурсу «/images» сервиса LUNA API:

```
"frame_store": "http://127.0.0.1:5000/6/images"
```

127.0.0.1 - IP адрес, где развернут сервис LUNA API;

1 - версия API сервиса LUNA API;

5000 - порт сервиса API по умолчанию.

См. дополнительную информацию о бакетах и ресурсе «/images» в руководстве администратора LUNA PLATFORM 5.

6.5.6 Группа параметров «authorization»

В данной группе параметров задаются либо token, либо account_id для выполнения запросов к сервису LUNA API.

Параметр event_handler > authorization > account_id должен совпадать с параметром account_id, задаваемом в запросе. Если поле авторизации не заполнено, то будет использован идентификатор account_id, задаваемый при создании потока (см. [Параметр «account_id»](#)).

См. подробную информацию об авторизации LUNA PLATFORM в руководстве администратора LUNA PLATFORM 5.

```
"authorization": {  
  "account_id": "557d54ec-29ad-4f3c-93b4-c9092ef12515"  
}
```

6.6 Группа параметров «policies»

6.6.1 Группа параметров «sending»

В данной группе параметров определяется период, в течении которого будет проводиться анализ кадров для выбора лучшего кадра, а также определяются все параметры, связанные с составлением коллекции лучших кадров.

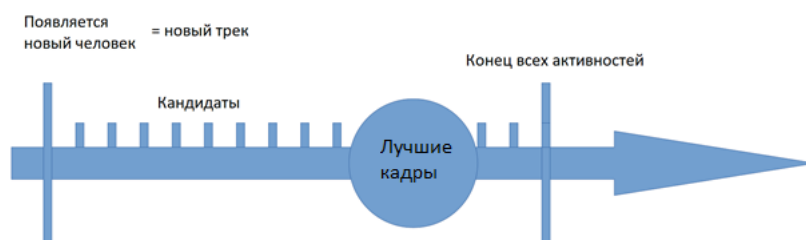


Рис. 10: Лучшие кадры

```
"sending": {  
  "time_period_of_searching": -1,  
  "silent_period" : 0,  
  "type" : "sec",  
  "number_of_bestshots_to_send": 1  
  "send_only_full_set": true  
  "delete_track_after_sending" false  
},
```

6.6.1.1 Параметр «time_period_of_searching»

Период анализа кадров, по истечении которого будет отправлен лучший кадр (период начинается с момента появления человека в кадре - первой детекции). Уменьшение этого параметра позволяет быстрее определять личность, но с большей погрешностью.

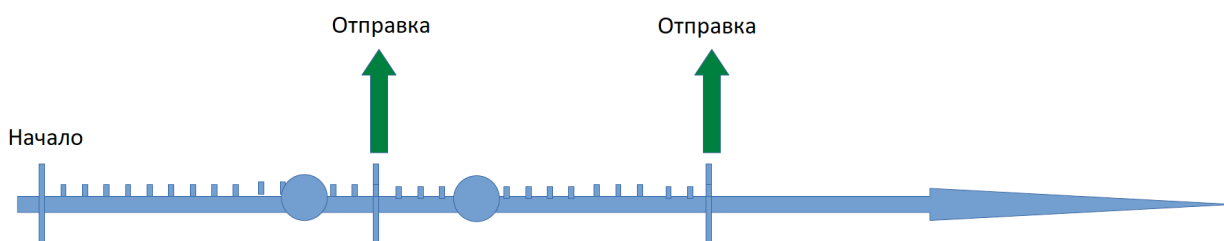


Рис. 11: Период отправки

Если значение равно «-1», то анализ кадров проводится по всем кадрам до конца трека. По оконч-

чании трека (когда объект покидает пределы кадра) лучший кадр будет отправлен во внешний сервис.

```
"time_period_of_searching": -1,
```

6.6.1.2 Параметр «silent_period»

Период ожидания, когда предыдущий анализ кадров завершился, а новый ещё не начался.

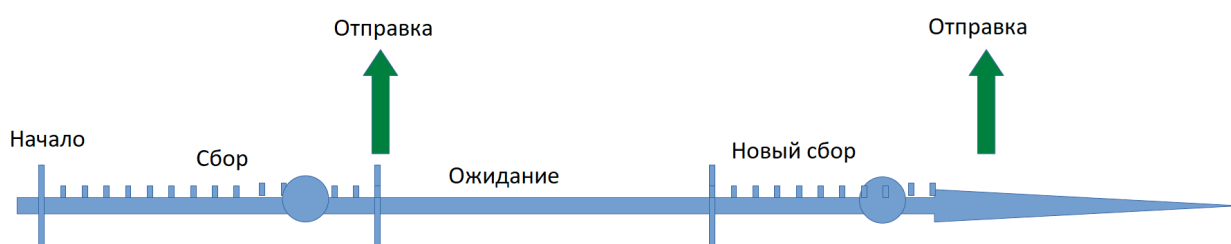


Рис. 12: Период ожидания

Если значение равно «-1», то период ожидания будет длиться бесконечно.

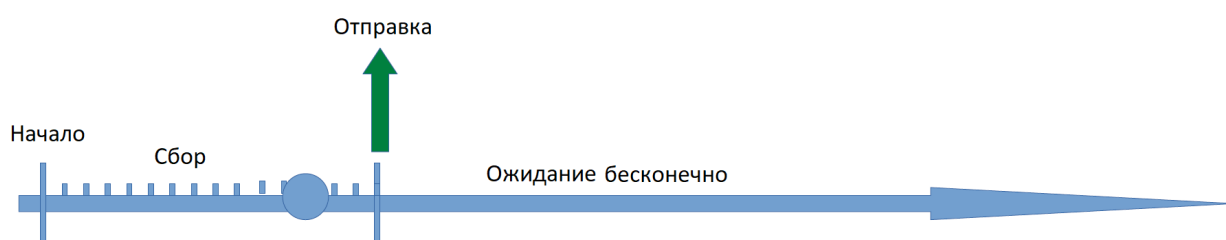


Рис. 13: Бесконечный период ожидания

```
"silent_period" : 0,
```

6.6.1.3 Параметр «type»

Параметр задаёт тип измерения периода анализа кадров и периода ожидания («frames» или «sec»).

```
"type" : "sec",
```

6.6.1.4 Параметр «number_of_bestshots_to_send»

Количество лучших снимков, которое пользователь хочет получить с трека или с какого-то промежутка времени на этом треке. Использование данного параметра предполагает создание коллекции из лучших снимков трека или временного отрезка трека, заданном в параметре «time_period_of_searching». Эта коллекция будет отправлена в LP.

```
"number_of_bestshots_to_send": 1
```

Увеличение значения повышает вероятность правильного распознавания объекта, но сказывается на загруженности сети.

6.6.1.5 Параметр «send_only_full_set»

Позволяет посылать данные только если имеется необходимое количество лучших снимков «number_of_bestshots_to_send» и длина трека «minimal_body_track_length_to_send».

```
"send_only_full_set" : true
```

6.6.1.6 Параметр «delete_track_after_sending»

Позволяет удалять трек после отправки.

```
"delete_track_after_sending" : false
```

6.6.2 Группа параметров «primary_track_policy»

Данная группа параметров используется только для работы с лицами.

Данная группа параметров предназначена для работы с Системами Контроля Управления Доступом (СКУД, турникеты на входах в банки/ офисные здания) для упрощения контроля и внедрения технологии распознавания лиц при входе на охраняемую территорию. Данная группа параметров **не используется** для типа «images».

6.6.2.1 Параметр «use_primary_track_policy»

Если значение этого параметра «true», то включается режим реализации Главного Трека (Primary Track). Логика работы заключается в следующем:

Из всех детекций на кадре выбирается максимальная по размеру детекция и её трек становится главным. На основе этого трека выполняется дальнейший анализ. Лучший кадр из этого трека отправляется во внешний сервис.

Все остальные треки обрабатываются в обычном режиме, но отправка лучшего кадра на внешний сервис происходит только для главного трека.

Как только другое лицо достигнет большего размера, чем лицо главного трека, трек этого лица станет новым главным треком и обработка будет выполняться для него.

При использовании параметра на контрольно-пропускном пункте будут отправляться лучшие снимки только того человека, который находится ближе всего к турникету (выполняется условие возникновения самой большой детекции).

```
"use_primary_track_policy": false,
```

6.6.2.2 Параметр «best_shot_min_size»

Параметр используется при включенном параметре «use_primary_track_policy».

Задаёт минимальный размер детекции, при котором начинается анализ кадров и определение лучшего кадра.

```
"best_shot_min_size": 70,
```

6.6.2.3 Параметр «best_shot_proper_size»

Параметр используется при включенном параметре «use_primary_track_policy».

Задаёт размер детекции для Главного Трека. Когда размер детекции достигает указанного значения, трек сразу отправляет все свои лучшие снимки в LP.

"best_shot_proper_size": 140

6.6.3 Группа параметров «liveness»

Данная группа параметров используется только для работы с лицами.

Liveness используется для проверки наличия живого человека в кадре и предотвращает использование распечатанного фото или фото с телефона для прохождения проверки.

Данную группу параметров рекомендуется использовать только по согласованию с сотрудниками VisionLabs.

Данная группа параметров не используется для типа «images».

6.6.3.1 Общие рекомендации по использованию Liveness

Liveness предназначен для применения только на автоматических контрольно-пропускных пунктах. Это сценарий, когда человек не задерживается перед камерой дольше десяти секунд.

Liveness позволяет минимизировать шанс прохода мошенника, использующего распечатанное фотоизображение другого человека или фотоизображение другого человека на телефоне.

В процессе работы Liveness возвращает значение, которое определяет степень уверенности системы в том, что в кадре живой человек. Значение находится в интервале от 0 до 1.

Требования к установке камеры

Для стабильной работы Liveness при установке камеры должны быть выполнены следующие условия:

- Лицо не должно выходить за пределы кадра. При этом расстояние от левого и правого краёв кадра должно быть больше или равно ширине найденного лица, а расстояние от верхнего и нижнего краёв кадра должно быть больше или равно высоте найденного лица;
- Кадр должен включать не только лицо, но и область груди;
- Камера должна быть расположена на уровне пояса и направлена на человека снизу вверх;
- Желательно, чтобы в кадр не попадали прямоугольные объекты (такие как оконные и дверные проёмы), обрамляющие область лица с четырёх сторон.

Пример правильного расположения камеры приведён на изображении ниже.

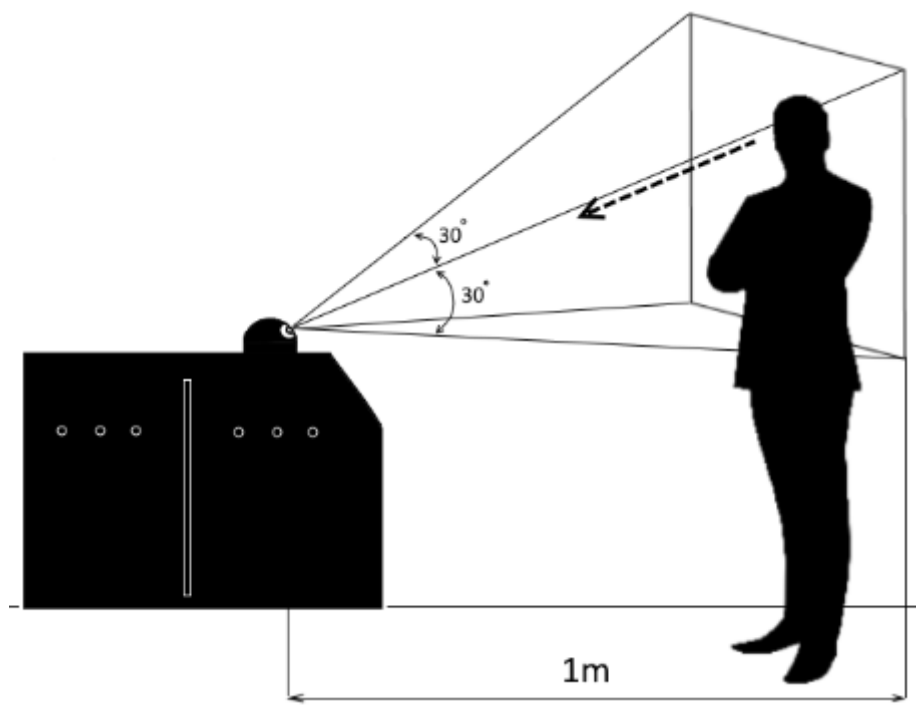


Рис. 14: Правильное расположение камеры для использования Liveness

При правильном расположении камеры уже на расстоянии в 3-4 метра FS начнёт собирать кадры и выбирать лучший снимок.

Посторонние объекты и люди, не проходящие через турникет, не попадают в зону видимости камеры.

FS выполнит отправку лучшего снимка, когда человек находится на расстоянии 1 метр от камеры. На этом расстоянии лицо достигнет необходимого размера для отправки.

Ниже приведён пример неправильного расположения камеры.

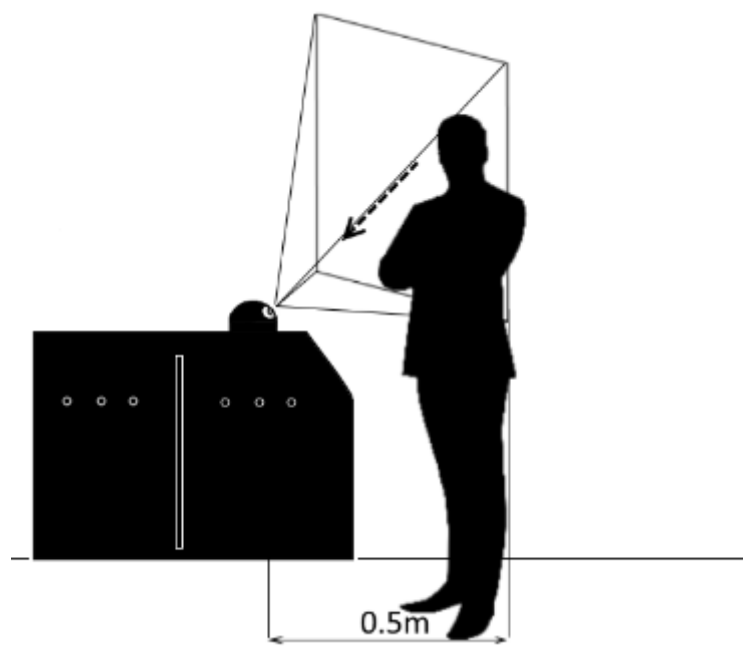


Рис. 15: Неправильное расположение камеры для использования Liveness

При неправильно настроенной камере:

- Человек попадает в кадр слишком поздно. FS не успевает получить необходимое количество кадров для обработки;
- Человек смотрит в камеру сверху вниз. Это ухудшает качество кадра для последующей обработки;
- В зону видимости камеры попадает пространство за пределами зоны интереса. В этом пространстве могут находиться люди или объекты, мешающие корректной работе FS.

Рекомендации по настройке FS

Рекомендуемые значения параметров секции «Liveness» приведены ниже.

```
"use_shoulders_liveness_filtration": true,  
"use_mask_liveness_filtration": true,  
"use_flying_faces_liveness_filtration": true,  
"liveness_mode": 1,  
"number_of_liveness_checks": 10,  
"liveness_threshold": 0.8,  
"livenesses_weights": [0.0, 0.25, 0.75],  
"mask_backgrounds_count": 300
```

Данные настройки не рекомендуется изменять.

Параметр `«best_shot_min_size»` следует устанавливать из расчета, что человек находится на расстоянии 3-4 метра до турникета.

Параметр `«best_shot_proper_size»` следует устанавливать из расчета, что человек находится на расстоянии 1 метр до турникета.

Для контроля выбора нужного человека используется параметр `«droi»`. Прямоугольник выбирается так, чтобы люди, имеющие намерение подойти к данному турникету, попали в прямоугольник как можно раньше. Это актуально для турникетов, расположенных близко к друг другу. Люди из соседних очередей могут попадать в зону видимости камер таких турникетов.

FAQ Liveness

При использовании Liveness обработка потока работает медленно

При разрешении камеры 1920x1080 и выше Mask Liveness работает медленно.

Для решения проблемы следует вручную уменьшить разрешение в камере до 720p. Это не скажется на качестве распознавания и работе Liveness, т.к. они работают без потери качества с лицами, размер которых примерно равен 100 пикселям.

Люди не могут пройти проверку Liveness при настройках FS по умолчанию

Возможные причины:

- Настройки в секции Liveness по умолчанию были изменены.

Не следует изменять настройки в секции Liveness, кроме настройки `«liveness_threshold»`.

Значение параметра `«liveness_threshold»` можно уменьшить, но оно не должно быть ниже «0.6».

- Liveness применяется не для целевого случая.

FS Liveness не предназначен для процессов авторизации и случаев долгого нахождения перед камерой.

- В зону видимости камеры попадают недопустимые объекты.

Например, если на заднем фоне находится экран, транслирующий видео, Liveness работать не будет.

- Для камеры задано неподходящее разрешение.

Проверьте разрешение камеры. См. «При использовании Liveness обработка потока работает медленно».

- Происходит задержка передачи кадров.

Если камера не передаёт кадры в реальном времени, то кадры могут поступать с задержкой.

- Значение `«best_shot_min_size»` задано неправильно.

При завышенном параметре `«best_shot_min_size»` Liveness не успевает накопить нужное число разнообразных кадров.

6.6.3.2 Параметр `«use_shoulders_liveness_filtration»`

Параметр включает режим проверки наличия человека в кадре, основанный на работе с областью головы и плеч.

```
"use_shoulders_liveness_filtration": false,
```

6.6.3.3 Параметр `«use_mask_liveness_filtration»`

Параметр включает режим проверки наличия человека в кадре, основанный на работе с фоном.

Скорость выполнения проверки зависит от размера кадров потока. Если при включённом параметре скорость обработки уменьшается, необходимо уменьшить разрешение видео в настройках камеры (например, до 1280x720).

```
"use_mask_liveness_filtration": false,
```

6.6.3.4 Параметр `«use_flying_faces_liveness_filtration»`

Параметр включает режим проверки наличия человека в кадре, основанный на работе с окружением лица.

```
"use_flying_faces_liveness_filtration": false,
```

6.6.3.5 Параметр `«liveness_mode»`

Параметр позволяет задать, для каких кадров трека будет проводиться проверка Liveness. Есть три варианта выбора кадров:

- 0 - Первые N кадров;
- 1 - Последние N кадров перед отправкой лучшего кадра (рекомендуемое значение);
- 2 - Все кадры трека.

Значение N задаётся в параметре `number_of_liveness_checks`.

```
"liveness_mode": 0,
```

6.6.3.6 Параметр «number_of_liveness_checks»

Параметр позволяет задать количество кадров в треке для проверки Liveness при использовании параметра *liveness_mode*.

Не рекомендуется выставлять значение меньше 10.

```
"number_of_liveness_checks": 10,
```

6.6.3.7 Параметр «liveness_threshold»

В данном параметре можно задать пороговое значение, при котором система будет считать, что в кадре живой человек. Система выдаст вердикт, что в кадре присутствует настоящий человек, только если Liveness вернул значение выше заданного порогового значения.

Рекомендуемое значение - «0.8». Не рекомендуется задавать значение ниже «0.6».

```
"liveness_threshold": 0.8,
```

6.6.3.8 Параметр «liveness_weights»

Параметр определяет коэффициент влияния проверки каждого типа (shoulders, mask и flying_faces) на итоговую оценку наличия живого человека в кадре.

Пользователь указывает три величины, относящихся к разным типам liveness в следующем порядке:

- Use_shoulders_liveness_filtration,
- Use_mask_liveness_filtration,
- Use_flying_faces_liveness_filtration.

Величины указываются в долях единицы. В приведенном ниже примере (являющимся настройкой по умолчанию) цифра 0.0 указывает, что 0% оценки будет основываться на shoulders_liveness, 0.25 - 25% на mask_liveness и 0.75 - 75% на flying_faces_liveness. Соотношение всегда масштабируется исходя из данных цифр вне зависимости от того составляют ли они единицу и какие из методов liveness включены.

```
"livenesses_weights": [0.0, 0.25, 0.75]
```

6.6.3.9 Параметр «mask_backgrounds_count»

Количество кадров фона, которые используются для соответствующей проверки.

Изменение данного параметра запрещено.

```
"mask_backgrounds_count": 300
```

6.6.4 Группа параметров «filtering»

Группа параметров описывает объекты фильтрации изображений и отправки результирующих портретов.

```
"filtering": {  
  "min_score": 0.5187,  
  "detection_yaw_threshold": 40,  
  "detection_pitch_threshold": 40,  
  "detection_roll_threshold": 30,  
  "yaw_number": 1,  
  "yaw_collection_mode": false,  
  "mouth_occlusion_threshold" : 0.0  
},
```

6.6.4.1 Параметр «min_score»

Min_score, также именуемый Approximate Garbage Score (AGS) для лиц и Detector score для тел – оценка приемлемости детекции, порог для фильтрации отправляемых детекций. Все детекции с оценкой выше значения параметра могут быть отправлены на сервер в виде HTTP-запроса, иначе детекции не считаются приемлемыми для дальнейшей работы с ними.

Если порог новой детекции выше имеющихся, детекция с наименьшим значением в коллекции будет заменена новой.

Рекомендуемое значение порога было выявлено путем проведения исследований и анализа детекций на различных изображениях лиц и тел.

```
"min_score" : 0.5187,
```

6.6.4.2 Параметр «detection_yaw_threshold»

Данный параметр используется только для работы с лицами.

Параметр задаёт максимальное значение угла поворота головы вправо и влево относительно камеры.

Если угол поворота головы на кадре больше заданного значения, то кадр считается неприемлемым для дальнейшей обработки.

```
"detection_yaw_threshold" : 40,
```


6.6.4.3 Параметр «detection_pitch_threshold»

Данный параметр используется только для работы с лицами.

Параметр задаёт максимальное значение угла наклона головы вниз и вверх относительно камеры.

Если угол наклона головы на кадре больше заданного значения, то кадр считается неприемлемым для дальнейшей обработки.

```
"detection_pitch_threshold" : 40,
```

6.6.4.4 Параметр «detection_roll_threshold»

Данный параметр используется только для работы с лицами.

Параметр задаёт максимальное значение угла наклона головы влево и вправо относительно камеры.

Если угол наклона головы на кадре больше заданного значения, то кадр считается неприемлемым для дальнейшей обработки.

```
"detection_roll_threshold" : 30,
```

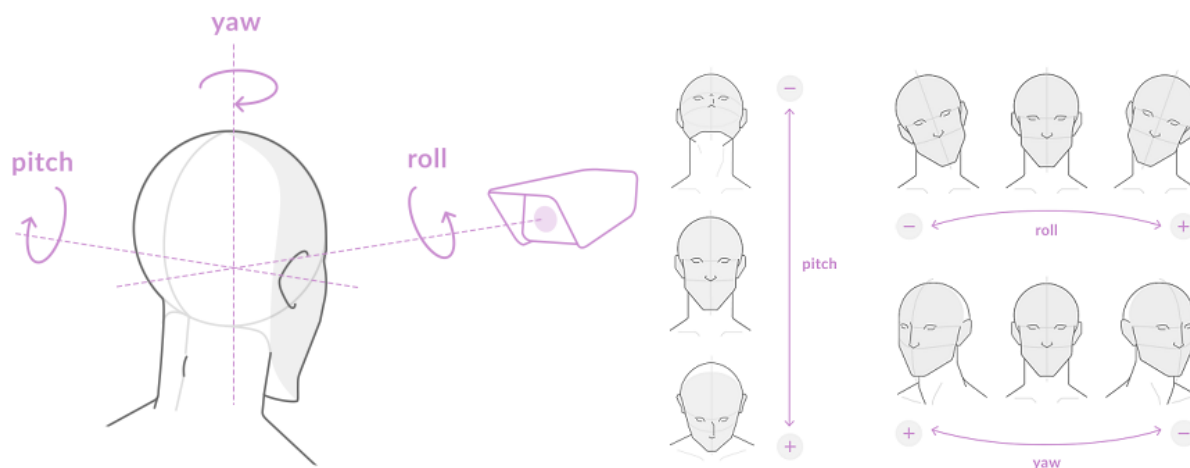


Рис. 16: Иллюстрация положения головы

6.6.4.5 Параметр «yaw_number»

Данный параметр используется только для работы с лицами.

Параметр определяет количество кадров, используемых для фильтрации фотоизображений по

углу поворота головы. Фильтрация отсекает изображения с сильно повернутыми от камеры лицами.

Как это работает:

В параметре задаётся количество кадров. Специальный алгоритм анализирует углы поворота головы на каждом из этих кадров. Если угол резко отличается от среднего значения в группе, то кадр не будет рассматриваться в качестве лучшего кадра.

Пример: Установим значение параметра «7», то есть будут рассмотрены 7 кадров. Представим, что на шести кадрах из семи посчитали, что угол поворота головы относительно камеры находится в диапазоне 50 - 60 градусов, а на седьмом кадре значение угла равно 0. Такое резкое изменение угла поворота головы за такой короткий промежуток времени невозможно, и, скорее всего, угол на седьмом кадре определен неверно. Седьмой кадр будет отсечен и не будет отправлен на внешний сервис в качестве лучшего кадра

По умолчанию параметр выключен, значение равно «1». Рекомендуемое значение – «7».

```
"yaw_number": 1,
```

6.6.4.6 Параметр «yaw_collection_mode»

Данный параметр используется только для работы с лицами.

Параметр указывает системе, что следует собрать некоторое количество кадров (yaw_number) для анализа углов поворота головы. Лучший кадр будет выбран среди этих кадров.

Если «yaw_collection_mode» отключен, то система будет последовательно проводить анализ поступающих кадров, т. е. сначала проводится анализ двух кадров, затем трех и т. д. Максимальное количество кадров этой последовательности задано в yaw_number.

По умолчанию параметр отключён.

```
"yaw_collection_mode" : false,
```

Цель работы параметров «yaw_number» и «yaw_collection_mode» — увеличить точность определения лучшего кадра из всего трека.

6.6.4.7 Параметр «mouth_occlusion_threshold»

Данный параметр используется только для работы с лицами.

Параметр определяет максимальную степень перекрытия рта у лица в кадре.

Например, при значении параметра равном «0.5» допускается перекрытие 50% области рта.

Если полученное значение превышает указанный порог, изображение считается неприемлемым для дальнейшей обработки.

Фильтрация проводится при значении параметра «0.3» и выше. При значении ниже фильтрация **отключена**.

```
"mouth_occlusion_threshold" : 0.0,
```

6.6.5 Параметр «frame_processing_mode»

Параметр используется только для типов «tcp», «udp» и «videofile».

Параметр аналогичен [convert_full_frame](#), но задаётся для конкретного экземпляра FaceStream.

При значении «full» кадр сразу конвертируется в RGB изображение необходимого размера после декодирования. При этом получается изображение лучшего качества, снижается скорость обработки кадров.

При значении «scale» масштабирование изображения выполняется в зависимости от настроек TrackeEngine (стандартное поведение для релизов 3.2.4 и ниже).

По умолчанию задано значение «auto». В этом случае один из двух режимов выбирается автоматически.

```
"frame_processing_mode": "auto",
```

6.6.6 Параметр «real_time_mode_fps»

Параметр используется только для типа «videofile».

Видео обрабатывается с заданным количеством FPS. Видео не может быть обработано с FPS выше, чем указано в данном параметре.

Если у видео высокое значение FPS и FaceStream не может работать с заданным количеством кадров в секунду, то кадры будут пропускаться.

Таким образом, видеофайл имитирует поток с реальной видеокамеры. Это может быть полезно при настройке производительности. Видео будет проигрываться с выбранной скоростью, что удобно для нагрузочного тестирования и последующего анализа.

Параметр не используется, если выставлено значение «0».

```
"real_time_mode_fps" : 0
```

6.6.7 Параметр «ffmpeg_threads_number»

Параметр позволяет задать количество потоков для декодирования видео с помощью FFMPEG.

При увеличении числа потоков увеличивается число ядер процессора, задействованных в декодировании. Увеличение числа потоков рекомендуется при обработке видео высокого разрешения (4K и выше).

```
"ffmpeg_threads_number" : 0
```

6.6.8 Группа параметров «health_check»

Данная группа параметров используется только при работе с потоками («tcp», «udp») и видеофайлами («videofile»).

В данной группе можно задать параметры повторного подключения к потоку при возникновении ошибок проигрывания потока.

6.6.8.1 Параметр «max_error_count»

Максимальное количество ошибок при воспроизведении потока.

Параметр работает совместно с параметрами «period» и «retry_delay». После получения первой ошибки, выполняется ожидание, заданное в параметре «retry_delay», а затем выполняется повторная попытка подключения к потоку. Если за время, указанное в параметре «period», было накоплено количество ошибок большее или равное количеству, указанному в «max_error_count», то обработка потока будет прекращена и его [статус](#) изменится на «failure».

Ошибки могут быть вызваны проблемой с сетью или доступностью видео.

```
"max_error_count": 10,
```

6.6.8.2 Параметр «period»

Задаёт время в секундах, за которое ведётся подсчёт ошибок.

Параметр работает совместно с параметрами «retry_delay» и «max_error_count». См. описание работы с данным параметром в разделе [«max_error_count»](#).

```
"period": 3600,
```

6.6.8.3 Параметр «retry_delay»

Задаёт время в секундах, по истечении которого выполняется повторная попытка подключения к потоку.

Параметр работает совместно с параметрами «period» и «max_error_count». См. описание работы с данным параметром в разделе [«max_error_count»](#).

```
"retry_delay": 5
```

6.7 Группа параметров «location»

Данная группа параметров включает информацию о расположении источника видеосигнала.

- «city» - город
- «area» - область
- «district» - район
- «street» - улица
- «house_number» - номер дома
- «geo_position» - географическое положение, включает в себя параметры «latitude» (широта) и «longitude» (долгота)

Географическое положение считается заданным правильно, если заданы долгота и широта.

Параметр `send_location_data` регулирует отправку расположения источника видеосигнала.

```
"location": {  
  "send_location_data" : false,  
  "city": "Moscow",  
  "area": "CAO",  
  "district": "Arbat",  
  "street": "Arbat",  
  "house_number": "37",  
  "geo_position": {  
    "longitude": 36.616,  
    "latitude": 55.752  
  }  
}
```

Этот параметр используется для создания событий в LUNA PLATFORM (см. документацию LUNA PLATFORM).

6.8 Группа параметров «autorestart»

Данная группа параметров позволяет настроить автоматический перезапуск потока. Доступно три параметра:

- `restart` - следует ли использовать автоматический перезапуск потока
- `attempt_count` - количество попыток автоматического перезапуска потока (по умолчанию 10)
- `delay` - задержка автоматического перезапуска потока, в секундах (по умолчанию 60 секунд)

```
"autorestart": {  
  "restart": 1,
```

```
"attempt_count": 7,  
"delay": 600  
}
```

6.9 Параметр «status»

Состояние на момент начала обработки. Доступно два состояния - «pending» (ожидание) «pause» (пауза).

```
"status": "pending"
```

Кроме двух состояний на момент начала обработки также доступны другие состояния, возникающие в процессе работы FaceStream (см. раздел [«Распределение потоков»](#)).

6.10 Параметры «group_name» и «group_id»

Параметры привязки потока к [группе](#). Можно указать либо «group_id», либо «group_name».

```
"group_id": "426542d6-5509-4e5b-8a01-e2abd5c0a8c6"
```

```
"group_name": "hd_stream_2"
```

7 Настройки, задаваемые в сервисе LUNA Configurator

В сервисе LUNA Configurator хранятся общие настройки для:

- FaceStream
- TrackEngine
- LUNA Streams

Для подробной информации о взаимодействии сервиса LUNA Configurator и FaceStream см. в разделе [«Использование FaceStream с LUNA Configurator»](#).

7.1 Настройки FaceStream

Конфигурирование параметров FaceStream осуществляется с помощью редактирования параметров в секции «FACE_STREAM_CONFIG» в сервисе Configurator (см. раздел [«Использование FaceStream с Configurator»](#));

Также можно настроить FaceStream с помощью редактирования конфигурационного файла «fs3Config.conf» в режиме работы без сервиса Configurator (см. раздел [«Использование FaceStream с конфигурационными файлами»](#)).

Далее приведены настройки, разбитые на логические блоки в зависимости от основных выполняемых блоком функций.

7.1.1 Группа параметров «logging»

Секция с настройками логирования работы приложения. Она отвечает за вывод сообщений об ошибках или о текущем состоянии приложения.

7.1.1.1 Параметр «severity»

Severity – параметр определяет информацию, которую пользователь хочет получать в логах. Доступны следующие фильтры информации:

0 – выводить всю информацию,

1 – выводить только предупреждения системы,

2 – выводить только ошибки.

```
"severity":  
{  
  "value": 1,  
  "description": "Logging severity levels ... "  
}
```


7.1.1.2 Параметр «tags»

Теги позволяют получить информацию о процессе обработки кадров и возникающих ошибках только для интересующих процессов FaceStream.

Данный параметр позволяет перечислить теги, которые связаны с логированием соответствующей информации. Если тег не указан, то соответствующая информация не логируется.

Информация по указанным тегам выводится в соответствии со значением параметра *severity*.

Текст логов содержит соответствующий тег. Его можно использовать для фильтрации логов.

Ошибки пишутся в лог всегда, для них не нужно задавать дополнительные теги.

Таблица 13: Описание тегов

Тег	Описание
streams	Информация, связанная с работой сервиса LUNA Streams
common	Общая информация
ffmpeg	Информация о работе библиотеки FFMPEG
liveness	Информация, связанная с проверкой наличия живого человека в кадре (секция «liveness»): достаточно ли информации о проверке liveness и проходит ли кадр проверку liveness
primary-track	Информация о главном треке (секция «primary_track_policy»): прошёл ли кадр указанные пороги и какой трек является главным
bestshot	Информация, связанная с выбором лучшего кадра: информация о появлении лучшего кадра, его смене, отправке лучшего кадра
angles	Информация о фильтрации по углам наклона головы
ags	Информация, связанная с оценкой качества кадров для дальнейшей обработки с помощью LUNA PLATFORM
mouth-occlusion	Информация о перекрытии лица в кадре
statistics	Информация о производительности, количестве обрабатываемых кадров в секунду и количестве пропущенных кадров
http_api	Информация об API запросах, отправленных в FaceStream, и получаемых ответах
client	Информация об отправке запросов в LUNA PLATFORM и полученных ответах
json	Информация об обработке параметров из конфигурационных файлов и сервиса Configurator

Тег	Описание
debug	Отладочная информация. Следует использовать данный тег только при отладке, а не при эксплуатации FS, т. к. выводится большое количество отладочной информации

```
"tags" : {
  "value" : ["common", "ffmpeg", "bestshot", "primary-track", "http_api",
    "client", "json", "streams"],
  "description" : "Logging specificity tags, full set: [streams, common,
    ffmpeg, liveness, primary-track, bestshot, angles, ags, mouth-
    occlusion, statistics, http_api, client, json, debug]"
},
```

7.1.1.3 Параметр «mode»

Mode – параметр задаёт режим логирования приложения: файл или консоль. Существует три режима:

- «l2c» – выводить информацию только в консоль,
- «l2f» – выводить информацию только в файл,
- «l2b» – выводить информацию в оба места.

```
"mode":
{
  "value": "l2b",
  "description": " The mode of logging ... "
}
```

В режиме работы FaceStream с конфигурационными файлами можно настроить директорию для сохранения логов при выводе информации в файл с помощью параметра запуска `--log-dir`.

7.1.2 Группа параметров «sending»

Данный раздел описывает настройку отправки готовых портретов в виде HTTP-запросов из FaceStream во внешние сервисы.

7.1.2.1 Параметр «request_type»

Данный параметр используется только для работы с лицами.

Данный параметр определяет тип запроса для отправки портретов во внешние сервисы. Поддерживаются 2 типа запросов:

- «jpeg» используется для отправки биометрических образцов в VisionLabs LUNA PLATFORM;
- «json» может использоваться для отправки портретов в сторонние пользовательские сервисы для дальнейшей обработки.

```
"request_type":
{
  "value": "jpeg",
  "description": " Type of request to server with portrait ..."
},
```

Подробное описание запросов см. в таблице.

Таблица 14: Запросы

Формат	Тип запроса	Заголовки авторизации	Тело
JSON	PUT	Authorization: Basic, login/password(Base64)	Media type: application/json; frame – исходный кадр в Base64 (если установлен параметр send_source_frame); data – портрет в Base64; identification – значение параметра Cid. Пример JSON: { frame": "", "data": "image_in_base_64", "identification": "camera_1" }
JPEG	POST	Authorization: Basic, login/password(Base64) или X-Auth-Token: 11c59254-e83f-41a3-b0eb-28fae998f271(UUID4)	Media type: image/jpeg

7.1.2.2 Параметр «portrait_type»

Данный параметр используется только для работы с лицами.

Данный параметр определяет формат, в котором изображение детектированного лица отправляется во внешний сервис. Возможные значения:

- «warp» – использовать биометрический образец, трансформированный под определённые требования;
- «gost» – не использовать трансформацию, вырезать детекцию из исходного кадра с учетом отступов.

Характеристики биометрического образца:

- имеет размер 250x250 пикселей;
- лицо отцентрировано на изображении;
- лицо повернуто так, чтобы условная линия, соединяющая уголки глаз, была приближена к горизонтальной.

Такой формат изображения при использовании совместно с LUNA PLATFORM дает следующие преимущества:

- постоянный предсказуемый минимальный объем данных для передачи по сети;
- фазы детектирования лица в LUNA PLATFORM автоматически отключаются для таких изображений, что приводит к значительному снижению времени обработки запроса.

```
"portrait_type":
{
  "value": "warp",
  "description": "Image format type..."
}
```

7.1.2.3 Параметр «send_source_frame»

Данный параметр позволяет отправить исходный кадр, на котором было детектировано лицо или тело.

При отправке изображения в LUNA PLATFORM нужно указать URL сервиса LUNA Image Store в параметре «[frame_store](#)».

Для коллекции лучших снимков («[number_of_bestshots_to_send](#)» > 1) отправляется только один исходный кадр, который определяется FaceStream как лучший из всех лучших снимков. Например, если значение настройки «[number_of_bestshots_to_send](#)» = 3, то в LP будет отправлено три лучших снимка и только один исходный кадр, который будет автоматически выбран из трех лучших снимков.

Исходный кадр отправляется в LP за некоторое время до отправки лучшего снимка. Он сохраняется в БД LUNA Events и ему назначается уникальный идентификатор, который хранится в поле «image_origin» таблицы «face_detect_result»/«body_detect_result». Далее в LP отправляется лучший снимок и генерируется событие, где в поле «image_origin» указывается ID исходного кадра, взятого из БД LUNA Events.

```
"send_source_frame" :
{
    "value": false,
    "description": "Send source frame for portrait from stream (false by
        default)."
```

7.1.2.4 Параметр «size_source_frame»

Параметр изменяет ширину исходного кадра до заданного значения. Допустимый диапазон - [0...1024]. Значение «0» означает, что ширина не будет меняться.

```
"size_source_frame" : {
    "value" : 0,
    "description" : "Resize source frame to given value. Range [0..1024]. ('
        0' by default, no resizing required)."
```

7.1.2.5 Параметр «detection_path_length»

Данный параметр используется только для работы с телами.

Данный параметр устанавливает максимальное количество детекций для параметра «minimal_body_track_length_to_send».

Максимальное количество детекций не может превышать 100. Если детекций больше 100, то алгоритм FaceStream удалит лишние детекции с некоторым шагом.

```
"detection_path_length" : {
    "value" : 100,
    "description" : "Maximum length of detection path allowed for sending to
        Luna ('100' by default)."
```

7.1.2.6 Параметр «minimal_body_track_length_to_send»

Данный параметр используется только для работы с телами.

Данный параметр позволяет вместе с лучшими снимками отправлять заданное количество детекций с координатами тела человека - x, y, width и height (см. запрос «save event» в документе OpenAPI LUNA PLATFORM). По наборам детекций тела человека можно определить его путь.

Данный параметр необходимо использовать совместно с параметром [detection_handler > handler_id](#). При использовании параметра, помимо генерации основного события, будет создаваться ещё одно событие, связанное с основным по «track_id». Данное событие будет содержать координаты тела человека и базовую информацию о событии (время детекции, время создания события, идентификатор аккаунта и т.п).

Детекция, у которой есть лучший снимок обязательна к отправке.

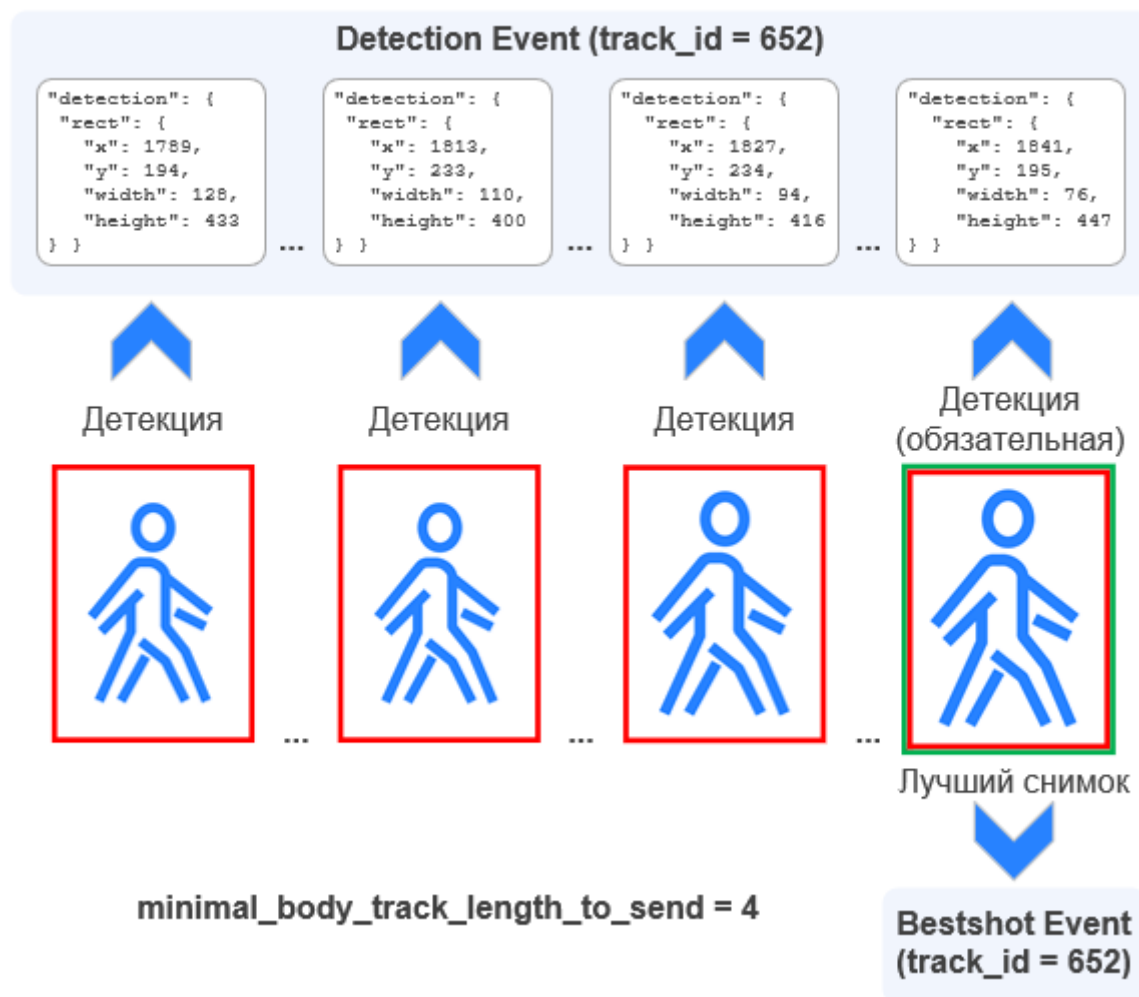


Рис. 17: Отправка детекций

Если количество детекций меньше заданного значения, то эти детекции не будут отправлены. Например, если количество детекций равно «3», а значение «minimal_body_track_length_to_send» равно «4», то детекции не будут отправлены и в логах FaceStream отобразится следующее сообщение:

```
Track is too short and will not be sent. Length 3 < 4
```

Если значение параметра равно «0», то отправка детекций выполняться не будет.

```
"minimal_body_track_length_to_send" : {  
    "value": 3,  
    "description" : "Minimal body track length to send"  
}
```

7.1.2.7 Параметр «[async_requests](#)»

Данный параметр позволяет указать, как следует выполнять запросы в LUNA PLATFORM – асинхронно или синхронно.

По умолчанию выставлен асинхронный режим, при котором все запросы в LUNA PLATFORM выполняются параллельно.

```
"async_requests" : {  
    "value" : true,  
    "description" : "Asynchronous requests to Luna server (true by default)."  
},
```

7.1.2.8 Параметр «[aggregate_attr_requests](#)»

Данный параметр позволяет включать агрегацию лучших кадров для получения одного биометрического шаблона в LUNA PLATFORM.

Агрегация выполняется, если отправляется больше одного лучшего кадра. Количество кадров для отправки задаётся параметром «[number_of_bestshots_to_send](#)».

Точность распознавания лиц и тел при использовании агрегированного биометрического шаблона выше.

```
"aggregate_attr_requests" :  
{  
    "value" : true,  
    "description" : "Set aggregate attributes in request to luna api 6 if  
        there are more than one bestshot (true by default)."  
},
```

7.1.2.9 Параметр «jpeg_quality_level»

Качество JPEG для отправки исходных кадров:

- «best» - сжатие изображения не выполняется
- «good» - 75% исходного качества изображения
- «normal» - 50% исходного качества изображения
- «average» - 25% исходного качества изображения
- «bad» - 10% исходного качества изображения

Качество «best» выставлено по умолчанию.

Отправка изображений в высоком качестве может влиять на скорость обработки кадров.

```
"jpeg_quality_level" : {  
  "value" : "best",  
  "description" : "Level of jpeg quality for source frames ['best', 'good',  
    'normal', 'average', 'bad'] ('best' by default)."  
}
```

7.1.3 Группа параметров «lunastreams»

Данный раздел описывает настройку отправки готовых портретов в виде HTTP-запросов из FaceStream в сервис LUNA Streams.

См. подробную информацию о работе LUNA Streams с FaceStream в разделе [«Взаимодействие FaceStream с LUNA Streams»](#).

7.1.3.1 Параметр «origin»

Адрес и порт сервера, где запущен сервис LUNA Streams.

```
"origin": {  
  "value": "http://127.0.0.1:5160",  
  "description": "LunaStreams url address."  
}
```

7.1.3.2 Параметр «api_version»

В параметре задаётся версия API сервиса LUNA Streams. На настоящий момент поддерживается 1 версия API.


```
"api_version": {  
  "value": 1,  
  "description": "Api version."  
}
```

Актуальную версию API всегда можно найти в документации сервиса API.

7.1.3.3 Параметр «max_number_streams»

В параметре задаётся верхняя граница количества потоков FS. Значение должно быть больше 0.

```
"max_number_streams": {  
  "value": 50,  
  "description": "Upper bound on the number of streams FS processes. Value  
    must be greater than 0."  
}
```

7.1.3.4 Параметр «request_stream_period»

В параметре задаётся временной промежуток между запросами на получение новых потоков из LUNA Streams в диапазоне от 0.1 до 3600 секунд.

Значение по умолчанию - 1 секунда.

```
"request_stream_period": {  
  "value": 1.0,  
  "description": "Time period for requesting new streams from LUNA Streams  
    . Available range [0.1, 3600]. Default value 1 second."
```

7.1.3.5 Параметр «send_feedback_period»

В параметре задаётся временной промежуток между отправкой отчётов об обработанных потоках в LUNA Streams в диапазоне от 1.0 до 3600 секунд.

Значение по умолчанию - 5 секунд.

Значение данного параметра не должно превышать значение параметра «[STREAM_STATUS_OBSOLETEING_PERIOD](#)», задаваемого в настройках сервиса LUNA Streams.

```
"send_feedback_period": {  
  "value": 5.0,
```

```
"description": "Time period for sending report of streams. Available  
range [1.0, 3600]. Default value 5 seconds. Must not be larger than  
STREAM_STATUS_OBSOLETING_PERIOD in LUNA Streams."  
}
```

7.1.3.6 Параметр «max_feedback_delay»

Параметр задаёт максимальную задержку отправки отчёта в диапазоне от 1.0 до 3600 секунд. Если отчёт не был отправлен за данное время, то FaceStream остановит обработку текущих потоков.

Значение данного параметра не должно быть меньше, чем значение параметра «send_feedback_period» и не должно превышать значение параметра «STREAM_STATUS_OBSOLETING_PERIOD», задаваемого в настройках сервиса LUNA Streams.

Значение по умолчанию - 10 секунд.

```
"max_feedback_delay": {  
  "value" : 10.0,  
  "description": "Max feedback sending delay after which processing streams  
stops. Available range [1.0, 3600]. Default value 10 seconds. Must  
not be less than send_feedback_period and larger than  
STREAM_STATUS_OBSOLETING_PERIOD in LUNA Streams."  
}
```

7.1.4 Группа параметров «performance»

7.1.4.1 Параметр «stream_images_buffer_max_size»

Параметр задаёт максимальный размер буфера изображений для одного потока.

При увеличении значения параметра повышается производительность FaceStream. Чем выше заданное значение, тем больше памяти требуется для работы.

Параметр рекомендуется выставить равным 40 при работе с GPU, если достаточно памяти GPU.

```
"stream_images_buffer_max_size" : {  
  "value" : 40,  
  "description" : "Max images buffer size for a single stream. Higher value  
provides better perfomance, but increases memory consumption. When  
set to 0 buffer is not used. (40 by default)"  
}
```

7.1.4.2 Параметр «enable_gpu_processing»

Параметр позволяет включить использование GPU вместо CPU для вычислений.

Использование GPU позволяет ускорить вычисления, но при этом увеличивается потребление оперативной памяти.

Вычисления с использованием видеокарты поддерживаются только для детектора FaceDetV3. См. параметр «defaultDetectorType» в настройках FaceEngine («faceengine.conf»).

```
"enable_gpu_processing" : {
    "value" : false,
    "description" : "When 'true' the processing is performed using
        GPU instead of CPU. GPU could provide better performance, but
        increases memory consumption. ('false' by default)"
},
```

7.1.4.3 Параметр «convert_full_frame»

Если данный параметр включён, то кадр сразу конвертируется в RGB изображение необходимого размера после декодирования. При этом получается изображение лучшего качества, но снижается скорость обработки кадров.

Если параметр выключен, то масштабирование изображения выполняется в зависимости от настроек TrackEngine (стандартное поведение для релизов 3.2.4 и более ранних).

Параметр аналогичен параметру [frame_processing_mode](#), но задаётся сразу для всех экземпляров FaceStream.

```
"convert_full_frame" : {
    "value" : true,
    "description" : "Enables converting full raw frame from decoder
        to rgb for processing. If value is 'true', then better
        quality is achieved. 'false' value provides better performance
        . ('true' by default)"
}
```

7.1.5 Группа параметров «debug»

Секция **debug** используется для настройки и отладки приложения. Параметры данной секции не рекомендуется использовать в промышленной среде, поскольку они потребляют значительное количество ресурсов и снижают производительность.

7.1.5.1 Параметр «save_debug_info»

Параметр `save_debug_info` даёт возможность сохранять информацию о работе детектора и результатах распознавания. Если значение равно «true», то информация сохраняется и использу-

ется в отладочных целях для анализа качества работы системы.

```
"save_debug_info" :  
{  
  "value": false,  
  "description": "Save information for quality analysis ..."  
}
```

7.1.5.2 Параметр «save_only_jpegs_with_honest_detections»

Включение режима сохранения только кадров с детектированным лицом. Используется в отладочных целях для повторного покадрового анализа.

Данная настройка может значительно сэкономить пространство на жестком диске в случае, если лица появляются в кадре редко.

```
"save_only_jpegs_with_honest_detections" :  
{  
  "value": false,  
  "description": "Filter for save_jpegs flag to save only jpegs with  
    honest detections ('false' by default)."  
},
```

7.1.5.3 Параметр «save_jpegs»

Параметр save_jpegs позволяет сохранять кадры, поступившие на обработку в приложение. Используется в отладочных целях для повторного покадрового анализа.

Результатом сохранения станут кадры из исходного потока, что может потребовать значительного пространства на жёстком диске.

```
"save_jpegs" :  
{  
  "value": false,  
  "description": "Save jpegs for research visualization ..."  
}
```

7.2 Настройки TrackEngine

Данный раздел описывает параметры TrackEngine, которые используются для настройки FaceStream.

Конфигурирование параметров TrackEngine осуществляется с помощью редактирования параметров в секции «TRACK_ENGINE_CONFIG» в сервисе Configurator (см. раздел «[Использование FaceStream с Configurator](#)»).

Также можно настроить TrackEngine с помощью конфигурационного файла «trackengine.conf» в режиме работы без сервиса Configurator (см. раздел «[Использование FaceStream с конфигурационными файлами](#)»).

7.2.1 Параметры «use-face-detector» и «use-body-detector»

Данные параметры позволяют сменить детекцию лиц на тела и наоборот. Одновременная детекция лиц и тел невозможна.

Необходимо помнить, что для успешной смены детекции нужно задать соответствующие настройки FaceStream и настройки источников потоков.

```
<!-- use-face-detector: Flag to use or not face detection -->
<param name="use-face-detector" type="Value::Int1" x="1" />

<!-- use-body-detector: Flag to use or not body detection -->
<param name="use-body-detector" type="Value::Int1" x="0" />
```

7.2.2 Параметр «detector-step»

Параметр «detector-step» позволяет указать количество кадров, на которых будет выполняться редетекция лица или тела в заданной области до выполнения детекции лица или тела. Редетекция требует меньше ресурсов, но объект может быть потерян при задании большого количества кадров для редетекции.

```
<!-- detector-step: The count of frames between frames with full detection,
[0 .. 30] ('7' by default). -->
<param name="detector-step" type="Value::Int1" x="7" />
```

7.2.3 Параметр «detector-scaling»

Параметр «detector-scaling» позволяет масштабировать кадр перед обработкой.

```
<!-- detector-scaling: Scale frame before detection for performance reasons,  
      [0, 1] ('0' by default). -->  
<param name="detector-scaling" type="Value::Int1" x="0" />
```

7.2.4 Параметр «scale-result-size»

Параметр «scale-result-size» задаёт максимальный размер кадра после масштабирования по наибольшей из сторон кадра.

Если исходный кадр имел размер 1920 на 1080 и значение «scale-result-size» равно 640, то FaceStream будет обрабатывать кадр размером 640 на 360. Для подбора оптимального значения «scale-result-size» следует постепенно уменьшать масштаб кадра и проверять, происходит ли на кадре детектирование лиц или тел. Следует выставить минимальный размер изображения, при котором все объекты в интересующей области детектируются.

Если кадр был обрезан с помощью параметра [«roi»](#), то масштабирование будет применено к этому уменьшенному кадру. В этом случае значение «scale-result-size» следует задавать в зависимости от наибольшей стороны области ROI.

На изображениях ниже приведены примеры детекций для кадра разрешением 1920x1080 и для того же кадра после его масштабирования до разрешения 960x640. Шесть лиц найдено на изображении с разрешением 1920x1080.

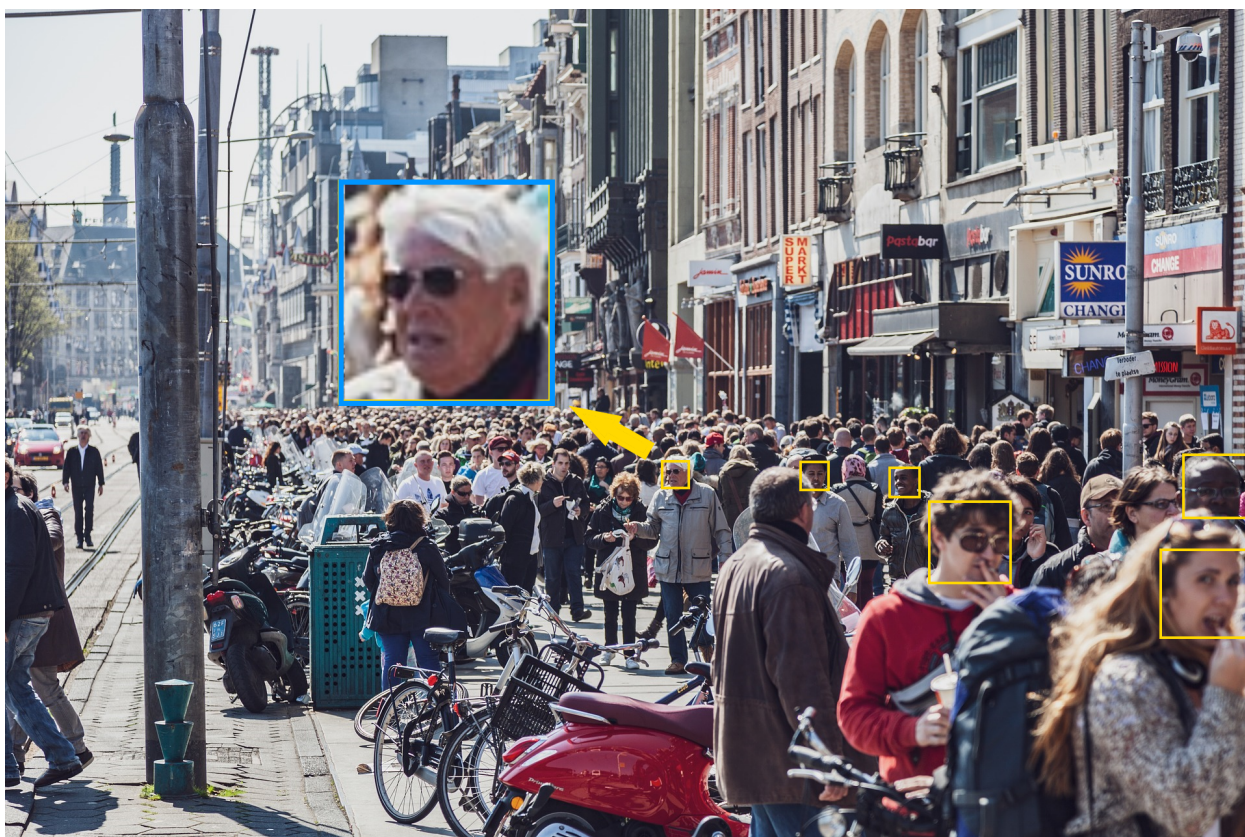


Рис. 18: Детекции на кадре с разрешением 1920X1080

Три лица найдено после масштабирования изображения до разрешения 960x640. Лица на заднем плане меньше по размеру и имеют низкое качество.

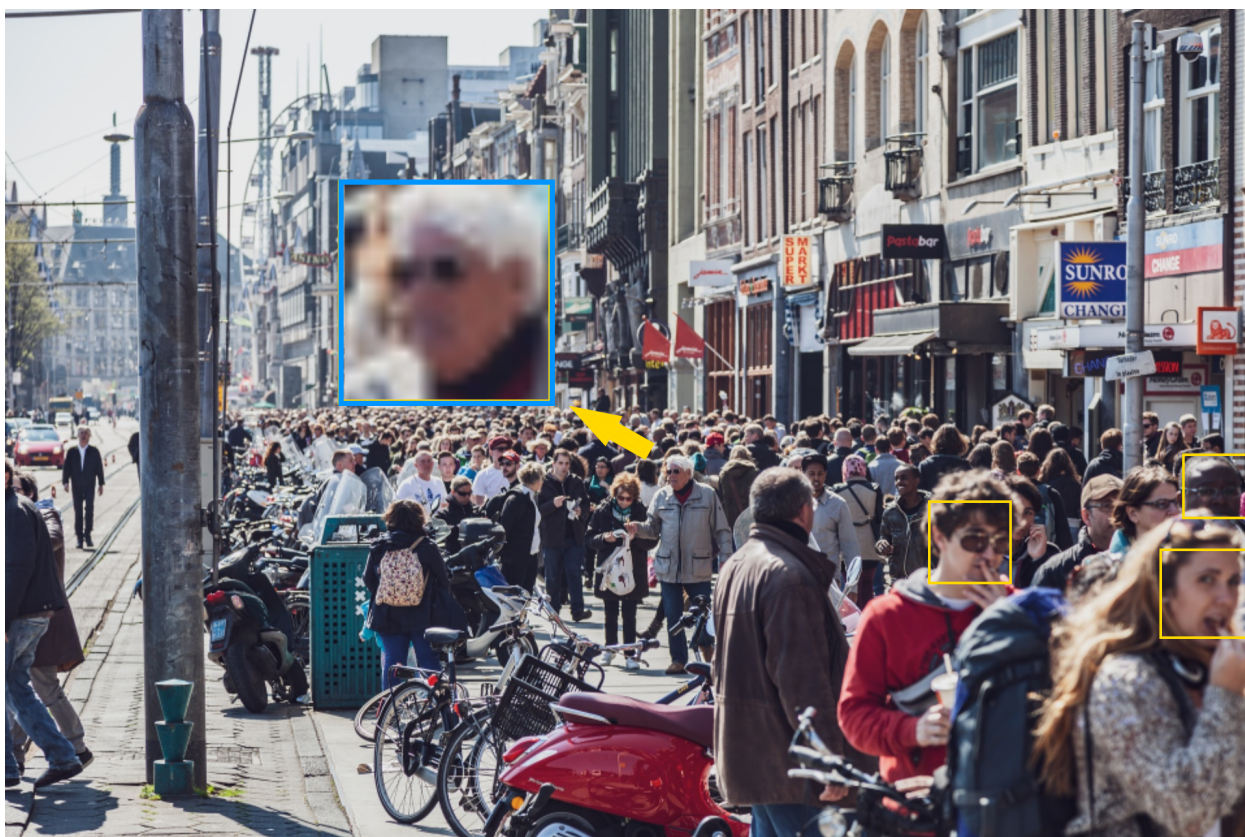


Рис. 19: Детекции на кадре с разрешением 960X640

Чем меньше разрешение кадра, тем меньше потребляется ресурсов.

При работе с телами данный параметр работает аналогичным образом.

```
<!-- scale-result-size: If scaling is enable, frame will be scaled to this
      size in pixels (by the max dimension - width or height).
      Upper scaling is not possible. ('640 by default') -->
<param name="scale-result-size" type="Value::Int1" x="640" />
```

7.2.5 Параметр «skip-frames»

Данный параметр используется только для работы с лицами.

Параметр «skip-frames» задаёт количество кадров, в течение которых система будет ожидать повторного появления объекта в области, где он исчез.

Обратите внимание, что слишком большие значения могут вести к уменьшению производительности.


```
<!-- skip-frames: If track wasn't updated by detect/redetect for this number
of frames, then track is finished ('36' by default). -->
<!-- note: very high values may lead to performance degradation. Parameter
doesn't affect on human tracking. -->
<param name="skip-frames" type="Value::Int1" x="36" />
```

7.2.6 Параметр «frg-subtractor»

При включении параметра «frg-subtractor» включается режим учёта перемещений в кадре. Последующая детекция лица или тела будет выполняться не на всём кадре, а только в областях с движением.

Определение областей с движением выполняется после масштабирования кадра.

Включение параметра «frg-subtractor» увеличивает производительность работы FaceStream.

```
<!-- frg-subtractor: Use foreground subtractor for filter of frames, [0, 1]
('1' by default). -->
<param name="frg-subtractor" type="Value::Int1" x="1" />
```

7.2.7 Параметр «frg-regions-alignment»

Параметр «frg-regions-alignment» позволяет задать выравнивание для областей с движением.

```
<!-- frg-regions-alignment: frg regions alignment. Useful for detector
better batching utilization. -->
<!-- 0 or negative values mean using non aligned regions, (0 by default).
-->
<param name="frg-regions-alignment" type="Value::Int1" x="0" />
```

7.2.8 Параметр «frg-regions-square-alignment»

При включённом параметре «frg-regions-square-alignment» ширина и высота области с движением всегда будут равны.

```
<!-- align frg regions to rect with equal sides (max side choosen). See frg-
regions-alignment, [0, 1] ('1' by default). -->
<param name="frg-regions-square-alignment" type="Value::Int1" x="1" />
```

7.2.9 Параметр «batched-processing»

Параметр «batched-processing» включает пакетную обработку кадров.

При работе с несколькими камерами, с каждой камеры собирается по кадру. После чего происходит обработка данного пакета кадров.

Если параметр отключён, то кадры обрабатываются один за другим.

При использовании режима пакетной обработки увеличивается задержка перед обработкой, но при этом сама обработка выполняется быстрее.

Рекомендуется включать параметр и при использовании GPU, и при использовании CPU.

```
<!-- batched-processing: Process streams frames in batch or separately, [0, 1] ('1' by default). -->
<param name="batched-processing" type="Value::Int1" x="1" />
```

7.2.10 Параметр «min-frames-batch-size»

Параметр «min-frames-batch-size» задаёт минимальное количество кадров, которое следует накопить со всех камер перед обработкой.

Рекомендуется выставлять значение параметра «min-frames-batch-size» равным количеству подключённых потоков при использовании GPU.

Рекомендуется выставлять значение параметра «min-frames-batch-size» равным «2» при использовании CPU.

```
<!-- min-frames-batch-size: stream frames min batch size value to process, ('0' by default). -->
<!-- higher values lead to higher processing latency but increase throughput and device utilization. -->
<!-- zero/negative values disable this feature, so any stream frames will be processed if they are available -->
<!-- note: this parameter should be regulated with 'max-frames-batch-gather-timeout' (see below) -->
<param name="min-frames-batch-size" type="Value::Int1" x="0" />
```

7.2.11 Параметр «max-frames-batch-gather-timeout»

Параметр «max-frames-batch-gather-timeout» задаёт время между обработкой пакетов кадров.

Если обработка одного кадра укладывается в указанное время и остается дополнительный запас, то FaceStream ожидает дополнительные кадры, чтобы повысить утилизацию GPU.

Если «max-frames-batch-gather-timeout» равен 20 мс, то за это время обрабатывается предыдущий пакет и собирается новый пакет. Через 20 мс начинается следующая обработка, даже если не удалось собрать число кадров равное «min-frames-batch-size». Следующая обработка кадров не может начаться до окончания обработки предыдущего пакета кадров.

Если он равен «0», то нет задержки на наполнение пакета кадрами, а обработка происходит непрерывно и «min-frames-batch-size» игнорируется.

Рекомендуется выставить значение параметра «max-frames-batch-gather-timeout» равным «0» и при использовании GPU, и при использовании CPU.

```
<!-- max-frames-batch-gather-timeout: max available timeout to gather next
stream frames batch (see 'min-frames-batch-size') from last processing
begin time point (measured in ms), ('-1' by default). -->
<!-- negative values disable this feature (no timeout, so only stream frames
batches with size no less than 'min-frames-batch-size' value will be
processed) -->
<!-- note: this parameter is complementary to 'min-frames-batch-size' and
controls min average fps of stream frames batches processing -->
<param name="max-frames-batch-gather-timeout" type="Value::Int1" x="-1" />
```

7.3 Настройки LUNA Streams

Данный раздел описывает параметры сервиса LUNA Streams, задаваемые в сервисе LUNA Configurator.

7.3.1 Группа параметров «LUNA_STREAMS_DB»

Данная группа параметров задаёт настройки подключения к созданной базе данных LUNA Streams.

7.3.1.1 Параметр «db_type»

Параметр задаёт тип используемой базы данных. Доступно два варианта - «postgres» или «oracle».

По умолчанию используется тип «postgres».

```
"db_type": "postgres"
```

7.3.1.2 Параметр «db_user»

Параметр задаёт имя пользователя базы данных.

По умолчанию используется имя пользователя «luna».

```
"db_user": "luna"
```

7.3.1.3 Параметр «db_password»

Параметр задаёт пароль для базы данных.

По умолчанию используется пароль «luna».

```
"db_password": "luna"
```

7.3.1.4 Параметр «db_name»

Параметр задаёт имя базы данных для типа «postgres» и имя SID для типа «oracle», к которой необходимо подключиться.

По умолчанию используется имя «luna_streams».

```
"db_name": "luna_streams"
```

7.3.1.5 Параметр «db_host»

Параметр задаёт IP-адрес сервера с базой данных LUNA Streams.

По умолчанию используется адрес «127.0.0.1». Такой адрес означает, что будет использоваться база данных LUNA Streams, расположенная на сервере с LUNA Configurator. Если база данных находится на ином сервере, то в данном параметре нужно указать корректный IP-адрес сервера с базой данных.

```
"db_host": "127.0.0.1"
```

7.3.1.6 Параметр «db_port»

Параметр задаёт Listener порт базы данных LUNA Streams.

По умолчанию используется порт «5432» для «postgres» и «1521» для «oracle».

```
"db_port": 5432
```

7.3.1.7 Параметр «connection_pool_size»

Параметр задаёт размер пула соединений к БД.

По умолчанию используется значение «5».

```
"db_settings": {  
    "connection_pool_size": 5  
}
```

7.3.2 Группа параметров «LUNA_STREAMS_LOGGER»

Данная группа параметров задаёт настройки логирования сервиса LUNA Streams.

7.3.2.1 Параметр «log_level»

Параметр задаёт уровень отладочной печати, по приоритету: «ERROR», «WARNING», «INFO», «DEBUG».

```
"log_level": "INFO"
```

По умолчанию используется значение «INFO».

7.3.2.2 Параметр «folder_with_logs»

Параметр задаёт папку, в которой хранятся логи (относительный путь начинается с каталога с приложением).

По умолчанию используется значение «./».

```
"folder_with_logs": "./"
```

7.3.2.3 Параметр «log_time»

Параметр задаёт формат времени, используемый в записях лога: «LOCAL» или «UTC».

```
"log_time": "LOCAL"
```

По умолчанию используется значение «LOCAL».

7.3.2.4 Параметр «log_to_stdout»

Параметр позволяет отправлять лог на стандартный вывод (stdout). Доступно два варианта - «true» или «false».

```
"log_to_stdout": true
```

По умолчанию используется значение «true».

7.3.2.5 Параметр «log_to_file»

Параметр позволяет отправлять лог в файл. Доступно два варианта - «true» или «false».

```
"log_to_file": true
```

По умолчанию используется значение «true».

7.3.2.6 Параметр «multiline_stack_trace»

Параметр включает или отключает многострочные трейсы в логах. Доступно два варианта - «true» или «false».

```
"multiline_stack_trace": true
```

По умолчанию используется значение «true».

7.3.3 Группа параметров «LUNA_LICENSES_ADDRESS»

Данная группа параметров задаёт настройки подключения к сервису LUNA Licenses.

7.3.3.1 Параметр «origin»

Параметр задаёт протокол, IP адрес и порт сервиса LUNA Licenses. IP адрес «127.0.0.1» означает, что будет использоваться сервис LUNA Licenses, расположенный на сервере с LUNA Configurator. Если сервис находится на ином сервере, то в данном параметре нужно указать корректный IP адрес сервера с запущенным сервисом LUNA Licenses.

```
"origin": "http://127.0.0.1:5120",
```

По умолчанию используется значение «http://127.0.0.1:5120».

7.3.3.2 Параметр «api_version»

Параметр задаёт версию API сервиса LUNA Licenses. Доступная версия API - «1».

```
"api_version": 1
```

7.3.4 Параметр «STREAM_WORKER_ASYNC_LOCK_TIMEOUT»

Параметр задаёт тайм-аут экземпляра LUNA Streams для блокировки строки в таблице базы данных в секундах в диапазоне (0, 1]. Значение данной настройки необходимо увеличить если статусы потоков не обновляются, что может быть обусловлено медленным соединением от сервиса к базе данных.

По умолчанию используется значение «10».

7.3.5 Параметр «STREAM_STATUS_OBSOLETING_PERIOD»

Параметр задаёт период устаревания состояния потока в секундах в диапазоне (0, 86400]. За данный период времени обработчик FaceStream должен передать отчёт в LUNA Streams. В противном случае статус потока будет изменен на «restart», а запоздалый отчёт будет отвергнут.

По умолчанию используется значение «0.1».

7.3.6 Параметр «LUNA_STREAMS_ACTIVE_PLUGINS»

Параметр задаёт список активных плагинов (см. информацию о работе плагинов в руководстве администратора LUNA PLATFORM 5).

По умолчанию используется значение «[]».

7.3.7 Параметр «STORAGE_TIME»

Параметр задаёт формат времени, используемый для записей в базе данных: «LOCAL» или «UTC».

По умолчанию используется значение «LOCAL».

7.3.8 Группа параметров «INFLUX_MONITORING»

В данной группе параметров задаются настройки мониторинга сервисов LUNA PLATFORM.

7.3.8.1 Параметр «send_data_for_monitoring»

Параметр позволяет включить или отключить отправку данных для мониторинга в InfluxDB. Доступно два варианта - «0» или «1».

```
"send_data_for_monitoring": 1
```

По умолчанию используется значение «1».

7.3.8.2 Параметр «use_ssl»

Параметр позволяет использовать HTTPS для подключения к InfluxDB. Доступно два варианта - «0» или «1».

```
"use_ssl": 0
```

По умолчанию используется значение «0».

7.3.8.3 Параметр «flushing_period»

Параметр задаёт частоту отправки данных мониторинга в InfluxDB (в секундах).

```
"flushing_period": 1
```

По умолчанию используется 1 секунда.

7.3.8.4 Параметр «host»

Параметр задаёт IP-адрес сервера с InfluxDB 2.x.

По умолчанию используется адрес «127.0.0.1». Такой адрес означает, что будет использоваться InfluxDB 2.x, расположенный на сервере с LUNA Configurator. Если InfluxDB 2.x находится на ином сервере, то в данном параметре нужно указать корректный IP-адрес InfluxDB 2.x.

```
"host": "127.0.0.1"
```

7.3.8.5 Параметр «port»

Параметр задаёт порт InfluxDB 2.x.

```
"port": 8086
```

7.3.8.6 Параметр «bucket»

Параметр задаёт имя бакета InfluxDB 2.x.

```
"bucket": "luna_monitoring"
```

По умолчанию используется имя бакета «luna_monitoring».

7.3.8.7 Параметр «organization»

Параметр задаёт рабочую область InfluxDB 2.x.

```
"organization": "luna"
```

По умолчанию используется значение «luna».

7.3.8.8 Параметр «token»

Параметр задаёт токен аутентификации InfluxDB 2.x.

8 Использование FaceStream с LUNA Configurator

Сервис LUNA Configurator позволяет хранить настройки FaceStream, TrackEngine и LUNA Streams и передавать их запущенным экземплярам FaceStream.

Сервис LUNA Configurator также позволяет хранить настройки всех необходимых для запуска FaceStream сервисов LUNA PLATFORM. См. настройки сервисов LUNA PLATFORM в руководстве администратора LUNA PLATFORM 5.

После запуска, FaceStream работает с заданными в LUNA Configurator настройками и, по умолчанию, не запрашивает их до перезапуска. При необходимости можно включить проверку наличия изменений в настройках и автоматический перезапуск экземпляра(ов) FaceStream с помощью ключа запуска «CONFIG_RELOAD» и задать период получения параметров с помощью ключа запуска «PULLING_TIME».

См. описание ключей запуска в разделе «Ручной запуск FaceStream» > «Команды запуска контейнера FaceStream» > «Ключи запуска» руководства по установке FaceStream.

Если проверка наличия изменений в настройках отключена, то для применения изменённых в LUNA Configurator настроек экземпляра FaceStream, необходимо вручную перезапустить этот экземпляр FaceStream.

8.1 Особенности работы с LUNA Configurator

Если использование сервиса LUNA Configurator указано в строке запуска FaceStream, но сервис недоступен, либо заданные параметры некорректны, то FaceStream выдаст ошибку в лог и не будет запущен. При этом настройки FaceStream и TrackEngine будут браться из локальных файлов конфигурационных файлов, если они были прокинуты в контейнер.

Для LUNA Streams не предусматривается использование конфигурационного файла.

Если локальные конфигурационные файлы также отсутствуют, то будут взяты настройки по умолчанию прописанные в коде FaceStream.

8.2 Параметры в Configurator

LUNA Configurator включает записи с заданными параметрами.

Каждая запись в LUNA Configurator содержит имя, тег и само тело конфигурации. Запись соответствует одному из конфигурационных файлов.

Параметры в сервисе LUNA Configurator имеют такие же имена, как в конфигурационных файлах («fs3Config.conf», «trackengine.conf») и документации.

Таблица 15: Соответствие данных в LUNA Configurator и конфигурационных файлов поставки

Имя записи в LUNA Configurator	Связанный файл конфигурации	Описание
FACE_STREAM_CONFIG	fs3Config.conf	Основные параметры FaceStream
TRACK_ENGINE_CONFIG	trackengine.conf	Параметры детекции и трекинга лица или тела

8.3 Задание настроек для нескольких экземпляров FaceStream

Если к сервису Configurator подключён один экземпляр FaceStream, то он будет использовать настройки, которые загружены по умолчанию.

Если требуется использовать несколько экземпляров FaceStream с разными настройками, то для каждой такой настройки следует создать отдельную запись с уникальным тегом.

Тег служит уникальным идентификатором для записи и указывается в ключах запуска, описанных в руководстве по установке. Таким образом конкретный экземпляр FaceStream может получить свои уникальные настройки.

Следует выполнить следующие шаги:

- Скопируйте необходимую запись, например, «FACE_STREAM_CONFIG», нажав кнопку **Duplicate**.



Рис. 20: Копирование записи

- После этого задайте тег и измените значения параметров.

Create new setting

Limitation

FACE_STREAM_CONFIG

Description (str <= 128 chars)

Face stream configuration

Value (depends on schema)

```
{"logging":{"severity":1,"tags":["ffmpeg","gstreamer","bestshot","primary-track"],"mode":"I2b"},"sending":{"request-type":"jpeg","portrait-type":"warp","send-source-frame":false,"luna-api":3,"async-requests":true,"luna-account-id":""},"web_tasks":{"concurrent-max-count":3,"max-file-size":52428800},"performance":{"stream-images-buffer-max-size":10,"enable-gpu-processing":false,"convert-full-frame":true}}
```

Tags (str, separate by ',')

setting tags(separate by ',')

Cancel

Create

Рис. 21: Задание тега

Нельзя создавать теги для записей по умолчанию.

9 Использование FaceStream с конфигурационными файлами

При необходимости можно запускать FaceStream независимо от настроек «FACE_STREAM_CONFIG» и «TRACK_ENGINE_CONFIG» сервиса LUNA Configurator, используя настройки из конфигурационных файлов.

При таком варианте запуска подразумевается, что зависимые сервисы LUNA PLATFORM также будут запущены с конфигурационными файлами. Описание запуска сервисов LUNA PLATFORM с конфигурационными файлами не приведено в данной документации.

Запуск FaceStream с настройками из конфигурационных файлов осуществляется с использованием следующих конфигурационных файлов:

- fs3Config.conf (настройки аналогичны секции «FACE_STREAM_CONFIG» в LUNA Configurator)
- trackengine.conf (настройки аналогичны секции «TRACK_ENGINE_CONFIG» в LUNA Configurator)
- faceengine.conf

Следует предварительно задать все необходимые параметры в данных файлах перед запуском FaceStream.

Команда ручного запуска контейнера с использованием конфигурационных файлов будет отличаться от команды запуска с Configurator и будет выглядеть следующим образом:

```
docker run \
-v /var/lib/fs/fs-current/extras/conf/configs/fs3Config.conf:/srv/facestream
/data/fs3Config.conf \
-v /var/lib/fs/fs-current/extras/conf/configs/faceengine.conf:/srv/
facestream/data/faceengine.conf \
-v /var/lib/fs/fs-current/extras/conf/configs/trackengine.conf:/srv/
facestream/data/trackengine.conf \
-v /etc/localtime:/etc/localtime:ro \
--restart=always \
--detach=true \
--name=facestream \
--network=host \
--env=PORT=34569 \
--entrypoint /srv/facestream/FaceStream \
dockerhub.visionlabs.ru/luna/facestream:v.5.1.12 \
--config-path /srv/facestream/data/fs3Config.conf \
--data-dir /srv/facestream/data \
--log-dir /srv/facestream/logs \
--http-address http://0.0.0.0:34569
```

Файлы настроек находятся в комплекте поставки FaceStream в директории «extras/conf/configs/» и добавляются в контейнер при запуске следующими командами:

```
-v /var/lib/fs/fs-current/extras/conf/configs/fs3Config.conf:/srv/facestream
/data/fs3Config.conf \
-v /var/lib/fs/fs-current/extras/conf/configs/faceengine.conf:/srv/
facestream/data/faceengine.conf \
-v /var/lib/fs/fs-current/extras/conf/configs/trackengine.conf:/srv/
facestream/data/trackengine.conf \
```

9.1 Ключи запуска с конфигурационными файлами

Для запуска FaceStream с конфигурационными файлами внутри контейнера используется следующая команда, которая позволяет указать корректные пути до директорий внутри контейнера.

```
--config-path /srv/facestream/data/fs3Config.conf \
--data-dir /srv/facestream/data \
--log-dir /srv/facestream/logs \
--streams-id 426542d6-5509-4e5b-8a01-e2abd5c0a8c6 ee4c42b6-23ae-410e-a2aa-
a4220e64ba4b
--groups-name stream_group-1 stream_group-2
--config-reload 1
--pulling-time 1800
--http-address http://0.0.0.0:34569
```

Доступны следующие ключи:

- `--help` - выдает перечень доступных ключей и их описание.
- `--config-path` - полный путь к файлу настроек приложения «fs3Config.cfg». Если определен данный параметр, то при поиске конфигурационного файла игнорируется путь к данным.
- `--data-dir` - путь к директории с данными детекторов и настройками.
- `--log-dir` - директория для записи файлов логирования.
- `--streams-id` - в теге задается список идентификаторов потоков, которые будут запрошены из LUNA Streams для обработки. Остальные потоки будут отфильтрованы. Параметр «stream_id» выдается в ответе на запрос «create stream».

Если тег `--streams-id` не задан, то FaceStream будет брать из очереди все существующие «stream_id».

Если задано несуществующее значение, то при запуске FaceStream будет указана ошибка о некорректном UUID.

- `--streams-name` - в теге задается список имен потоков. Имена потоков задаются с помощью параметра «name» во время их создания (запрос «create streams»). Потоки с данными именами будут запрошены из LUNA Streams для обработки. Остальные потоки будут отфильтрованы.

В остальном принцип работы схож с тегом `--streams-id`.

- `--groups-id` и `--groups-name` - в тегах задаются список идентификаторов групп или список имён групп. Параметры «group_id» или «group_name» задаются во время создания потока (запрос «create stream»). Потоки с данными параметрами будут запрошены из LUNA Streams для обработки. Остальные потоки будут отфильтрованы.

Если теги `--groups-id`/`--groups-name` не заданы, то FaceStream не будет фильтровать потоки по группам.

Если задано несуществующее значение, то при запуске FaceStream будет указана ошибка о некорректном UUID.

- `--config-reload` - тег, включающий проверку наличия изменений в файле «fs3Config.conf» и принимающий следующие значения:
 - «1» - отслеживание изменений включено, при наличии изменений в конфигурации будут автоматически перезапущены все контейнеры FaceStream.
 - «0» - отслеживание изменений отключено.

По умолчанию значение равно «1»

- `--pulling-time` - тег, задающий период получения новых параметров из файла «fs3Config.conf» в диапазоне [1...3600] сек. Используется совместно с тегом `--config-reload`.

По умолчанию значение равно «10».

- `--http-address` - HTTP адрес, который будет прослушивать FaceStream. Задаётся в формате «address:port» (используется только для FaceStream в серверном режиме). На этот адрес пользователь будет отправлять запросы.

Следует задать внешний IP сервера FaceStream. По умолчанию задано «http://0.0.0.0:34569».

10 Вывод информации в логи

В данном разделе описана дополнительная информация, которая может помочь при работе с логами FaceStream или логами требуемых для него сервисов.

10.1 Формат вывода логов FaceStream

Логи FaceStream имеют следующий формат:

```
[I0317 16:27:07.375125 57 LunaBaseClient.cpp:45] [client] Request
```

Здесь:

- I0317:
 - I - уровень логирования. В логах может отображаться 4 уровня - I (Info), W (Warning), E (Error), F (Fatal). При необходимости можно настроить уровень логирования (см. [параметр «severity»](#));
 - 0317 - день и месяц, т.е. 17 марта.
- 16:27:07.375125 - временная метка.
- 57 - PID идентификатор процесса.
- LunaBaseClient.cpp - имя файла, из-за которого появилась данная строка лога.
- 45 - номер строки.
- [client] - тег, связанный с логированием соответствующей информации (см. [параметр «tags»](#)).
- Request - описание строки лога.

Перечень ошибок FaceStream не представлен в данной главе.

10.2 Ошибки сервиса LUNA Streams

Данный раздел описывает ошибки, которые возвращает сервис LUNA Streams. Каждая из ошибок API имеет свой уникальный номер. Его удобно использовать, чтобы найти ошибку.

Некоторые из этих ошибок могут иметь несколько различных причин возникновения.

В случае возникновения «Internal server error» или иной непредвиденной ошибки следует обратиться к логам сервиса для получения дополнительной информации о проблеме.

При использовании LUNA Streams вместе с сервисами LUNA PLATFORM, могут возникнуть ошибки других сервисов LP. В таком случае следует обратиться к документации LUNA PLATFORM или посе-

тить сайт с [онлайн-документацией](#), содержащей полный перечень ошибок, возвращаемых сервисами LUNA PLATFORM 5.

10.2.1 Вернулся код ошибки 39001

Сообщение ошибки:

Object not found Stream with id {value} not found

Источник ошибки:

Ошибки сервиса LUNA Streams

Описание ошибки:

Поток с указанным идентификатором не был найден. Убедитесь, что задан существующий «stream_id». Получить список существующих «stream_id» можно с помощью запроса GET к ресурсу «/streams».

10.2.2 Вернулся код ошибки 39002

Сообщение ошибки:

Bad input data «{value}» is not valid stream status; permitted: {value}.

Источник ошибки:

Ошибки сервиса LUNA Streams

Описание ошибки:

При создании потока был введен некорректный статус в поле «status». Можно задавать только два статуса - «pause» и «pending». Остальные статусы можно получить только в определенный момент времени (см. раздел «[Таблица переходов от статуса к статусу](#)»). В описании ошибки приводится ожидаемый статус.

10.2.3 Вернулся код ошибки 39003

Сообщение ошибки:

Unable to stop processing Processing of stream with id «{value}» is already in progress and cannot be stopped.

Источник ошибки:

Ошибки сервиса LUNA Streams

Описание ошибки:

Невозможно выставить статус «pause» для указанного «stream_id», поскольку обработка уже началась (актуально только для видеофайлов).

10.2.4 Вернулся код ошибки 39004

Сообщение ошибки:

Bad input data «{value}» is not valid stream log target; permitted: {}.

Источник ошибки:

Ошибки сервиса LUNA Streams

Описание ошибки:

Указано несуществующее значение параметра «targets» в запросе «/streams/logs» на получение логов.

10.2.5 Вернулся код ошибки 39005

Сообщение ошибки:

Unable to cancel processing Processing of stream with id «{value}» is finished and cannot be cancelled

Источник ошибки:

Ошибки сервиса LUNA Streams

Описание ошибки:

Невозможно выставить статус «cancel» для указанного «stream_id», т.к. обработка уже завершена.

10.2.6 Вернулся код ошибки 39006

Сообщение ошибки:

Unique constraint error Group named {value} already exists

Источник ошибки:

Ошибки сервиса LUNA Streams

Описание ошибки:

Группа с указанным названием уже существует.

Укажите другое название или удалите существующую группу с помощью запроса «remove group».

10.2.7 Вернулся код ошибки 39007

Сообщение ошибки:

Object not found Group named {value} not found

Источник ошибки:

Ошибки сервиса LUNA Streams

Описание ошибки:

Не найдена группа с указанным именем.

Проверьте введенное имя группы.

Можно получить список всех существующих групп с их параметрами с помощью запроса «get groups».

10.2.8 Вернулся код ошибки 39008

Сообщение ошибки:

Object not found Group with id {value} not found”

Источник ошибки:

Ошибки сервиса LUNA Streams

Описание ошибки:

Не найдена группа с указанным идентификатором.

Проверьте введенный идентификатор группы.

Можно получить список всех существующих групп с их параметрами с помощью запроса «get groups».

10.2.9 Вернулся код ошибки 39009

Сообщение ошибки:

Object not found. Not found «{value}» preview url for stream with id «{value}».

Источник ошибки:

Ошибки сервиса LUNA Streams

Описание ошибки:

Не найден указанный url предварительного просмотра (live или last_frame) для указанного stream_id.

Убедитесь, что отчет содержит требуемый url-адрес.

10.2.10 Вернулся код ошибки 39010

Сообщение ошибки:

Preview processing error, {value}

Источник ошибки:

Ошибки сервиса LUNA Streams

Описание ошибки:

Ошибка обработки предварительного просмотра.

В скобках указывается ошибка, передаваемая от источника потока.

11 Совместимость с камерами

В таблице приведена информация о совместимости FaceStream с указанными IP камерами.

Таблица 16: Совместимость с камерами

Модель камеры	Тестирование	Версия FaceStream	Результаты тестирования
Hikvision ds-2cd2822f	Внутреннее тестирование	3.2.2	Проблем не выявлено
Hikvision DS-2CD7126G0-IZS	Информация от партнёров	3.2.2	Проблем не выявлено
Dahua IPC-HDBW8630E-Z	Информация от партнёров	3.2.2	Проблем не выявлено
WiseNet XNV 8040 WiseNet XNV 8030	Информация от партнёров	3.2.2	Проблем не выявлено
AXIS Q3515-LV	Информация от партнёров	3.2.2	Проблем не выявлено
Hikvision DS-2CD5126	Информация от партнёров	3.2.2	Проблем не выявлено
Vivotek cc8160	Информация от партнёров	3.2.2	Проблем не выявлено
Vivotek cc837	Информация от партнёров	3.2.2	Проблем не выявлено
ACTi E38	Информация от партнёров	3.2.2	Проблем не выявлено
ACTi A92	Информация от партнёров	3.2.2	Проблем не выявлено
ACTi E928	Информация от партнёров	3.2.2	Проблем не выявлено
Vivotek FD9365-EHTV	Информация от партнёров	3.2.2	Проблем не выявлено
Vivotek IB9367-EHT	Информация от партнёров	3.2.2	Проблем не выявлено
Bosch NDI-4502-A	Информация от партнёров	3.2.2	Проблем не выявлено

Модель камеры	Тестирование	Версия FaceStream	Результаты тестирования
Bosch NBN-50022-V3 (Use fisheye Lens 4mp 2.7-12mm Lens(DH-PLZ1040-D FOC 201706050010))	Информация от партнёров	3.2.2	Проблем не выявлено
Bosch NUC-21012-F2	Информация от партнёров	3.2.2	Проблем не выявлено
Dahua ipc-hdbw2220rp-vfs	Информация от партнёров	3.2.2	Проблем не выявлено
Dahua ipc-hfw2221r-vfs-ire6	Информация от партнёров	3.2.2	Проблем не выявлено
Samsung XNV-8040RP	Информация от партнёров	3.2.2	Проблем не выявлено
Samsung SNV-SNV-6013P	Информация от партнёров	3.2.2	Проблем не выявлено
Hikvision DS-2CD2125FWD-IS	Информация от партнёров	3.2.2	Проблем не выявлено
Hikvision DS-2CD2025FWD-I	Информация от партнёров	3.2.2	Проблем не выявлено
Hikvision DS-2CD4525FWD-IZH	Информация от партнёров	3.2.2	Проблем не выявлено
Mobotix MX-S16B (Видеомодуль MX-O-SMA-S-6D041)	Внутреннее тестирование	3.2.3	Проблем не выявлено
Panasonic WV-S6131/WV-S6130	Внутреннее тестирование	3.2.3	Проблем не выявлено
Panasonic WV-SC588A	Внутреннее тестирование	3.2.3	Проблем не выявлено