

VisionLabs FaceStream

Руководство администратора

v.5.2.3

Содержание

Глоссарий	9
1 Введение	12
2 Обзор	13
2.1 Работа FaceStream с лицами и телами	15
2.1.1 Работа с лицами	15
2.1.2 Работа с телами	17
2.2 Взаимодействие FaceStream с LUNA Streams	19
2.2.1 Диаграмма последовательности создания и обработки потока	19
2.2.2 Запросы к сервису LUNA Streams	22
2.2.3 Распределение потоков в LUNA Streams	23
2.2.3.1 Таблица переходов от статуса к статусу	23
2.2.4 Процесс обработки потоков в LUNA Streams	24
2.2.4.1 Автоматический перезапуск потоков	25
2.2.5 Автоматическое удаление логов потоков	26
2.2.6 Группировка потоков	27
2.2.7 Описание базы данных LUNA Streams	27
2.3 Рекомендации по настройке FaceStream	29
2.3.1 Рекомендации перед началом настройки	29
2.3.2 Настройка производительности FaceStream	29
2.3.2.1 Уменьшение области поиска лиц	30
2.3.2.2 Масштабирование кадра	32
2.3.3 Определение области с движением	32
2.3.4 Обработка пакетов кадров	33
2.3.5 Общая информация о настройке	34
2.3.5.1 Работа с треком	34
2.3.5.2 Отправка лучшего кадра	34
2.3.5.3 Фильтрация кадров	35
2.3.5.4 Работа со СКУД	36
2.4 Форматы, стандарты сжатия видео и протоколы	38
2.4.1 Форматы видео	38
2.4.2 Стандарты сжатия видео (кодеки)	38
2.4.3 Протоколы	38
2.5 Потребление памяти при работе FaceStream	40
2.6 Интерфейс воспроизведения потока	41
2.7 Совместный режим детектирования лиц и тел	41
2.7.1 Неподдерживаемый функционал	42

3	Перечень приоритетных настроек	43
3.1	Настройки для работы с лицами	44
3.2	Настройки для работы с телами	45
4	Настройки управления потоками	47
4.1	Общие настройки потока	48
4.2	Настройки обработки потока	49
4.2.1	type	50
4.2.2	reference	50
4.2.3	roi	51
4.2.4	frame_processing_mode	53
4.2.5	real_time_mode_fps	53
4.2.6	ffmpeg_threads_number	53
4.2.7	preferred_program_stream_frame_width	53
4.2.8	endless	54
4.2.9	mask	54
4.3	Настройки аналитики	56
4.3.1	Область выбора лучшего кадра	57
4.3.2	Фильтрация кадров	59
4.3.2.1	min_score_face	59
4.3.2.2	min_score_body	59
4.3.2.3	detection_yaw_threshold	59
4.3.2.4	detection_pitch_threshold	60
4.3.2.5	detection_roll_threshold	60
4.3.2.6	yaw_number	60
4.3.2.7	yaw_collection_mode	61
4.3.2.8	mouth_occlusion_threshold	61
4.3.2.9	min_body_size_threshold	62
4.3.3	Отправка во внешнюю систему	63
4.3.3.1	Настройки отправки данных тел	63
4.3.3.2	Настройки отправки лучших снимков	63
4.3.3.3	full_frame_settings	64
4.3.3.4	time_period_of_searching	64
4.3.3.5	silent_period	64
4.3.3.6	type	65
4.3.4	Настройки интеграции с внешней системой	66
4.3.4.1	frame_store	66
4.3.4.2	authorization	67
4.3.5	Настройки healthcheck	68
4.3.5.1	max_error_count	68

4.3.5.2	period	68
4.3.5.3	retry_delay	68
4.3.5.4	timeout	68
4.4	Настройки Liveness	69
4.4.1	Общие рекомендации по использованию Liveness	69
4.4.1.1	Требования к установке камеры	69
4.4.1.2	Рекомендации по настройке FS	71
4.4.1.3	FAQ Liveness	72
4.4.2	use_mask_liveness_filtration	73
4.4.3	use_flying_faces_liveness_filtration	73
4.4.4	liveness_mode	73
4.4.5	number_of_liveness_checks	73
4.4.6	liveness_threshold	73
4.4.7	livenesses_weights	74
4.4.8	mask_backgrounds_count	74
4.5	Настройки политики Главного трека	75
4.5.1	use_primary_track_policy	75
4.5.2	best_shot_min_size	75
4.5.3	best_shot_proper_size	76
5	Настройки, задаваемые в сервисе LUNA Configurator	77
5.1	Настройки FaceStream	77
5.1.1	logging	77
5.1.1.1	severity	77
5.1.1.2	tags	77
5.1.1.3	mode	79
5.1.2	sending	79
5.1.2.1	request_type	79
5.1.2.2	request_timeout_ms	80
5.1.2.3	portrait_type	80
5.1.2.4	crop_factor	82
5.1.2.5	max_crop_size	82
5.1.2.6	send_source_frame	82
5.1.2.7	size_source_frame	83
5.1.2.8	detection_path_length	84
5.1.2.9	minimal_body_track_length_to_send	84
5.1.2.10	async_requests	85
5.1.2.11	aggregate_attr_requests	85
5.1.2.12	jpeg_quality_level	85

5.1.3	lunastreams	85
5.1.3.1	origin	86
5.1.3.2	api_version	86
5.1.4	lunaplatform	86
5.1.4.1	origin	86
5.1.4.2	api_version	86
5.1.4.3	max_number_streams	86
5.1.4.4	request_stream_period	86
5.1.4.5	send_feedback_period	86
5.1.4.6	max_feedback_delay	87
5.1.5	performance	87
5.1.5.1	stream_images_buffer_max_size	87
5.1.5.2	enable_gpu_processing	87
5.1.5.3	convert_full_frame	87
5.1.6	fps_floor	88
5.1.7	monitoring	88
5.1.7.1	storage_type	88
5.1.7.2	send_data	88
5.1.7.3	organization	88
5.1.7.4	bucket	88
5.1.7.5	token	89
5.1.7.6	origin	89
5.1.7.7	flushing_period	89
5.2	Настройки TrackEngine	90
5.2.1	use-face-detector и use-body-detector	90
5.2.2	detector-step	90
5.2.3	detector-scaling	91
5.2.4	scale-result-size	91
5.2.5	skip-frames	93
5.2.6	frg-subtractor	94
5.2.7	frg-regions-alignment	94
5.2.8	frg-regions-square-alignment	94
5.2.9	batched-processing	95
5.2.10	min-frames-batch-size	95
5.2.11	max-frames-batch-gather-timeout	95
5.3	Настройки LUNA Streams	97
5.3.1	LUNA_STREAMS_DB	97
5.3.1.1	db_type	97
5.3.1.2	db_host	97

5.3.1.3	db_port	97
5.3.1.4	db_user	97
5.3.1.5	db_password	97
5.3.1.6	db_name	98
5.3.1.7	db_settings> connection_pool_size	98
5.3.1.8	dsn	98
5.3.2	LUNA_LICENSES_ADDRESS	99
5.3.2.1	origin	99
5.3.2.2	api_version	99
5.3.3	LUNA_STREAMS_LOGGER	99
5.3.3.1	log_level	99
5.3.3.2	log_time	100
5.3.3.3	log_to_stdout	100
5.3.3.4	log_to_file	100
5.3.3.5	folder_with_logs	100
5.3.3.6	max_log_file_size	101
5.3.3.7	multiline_stack_trace	101
5.3.3.8	format	101
5.3.4	LUNA_MONITORING	101
5.3.4.1	storage_type	102
5.3.4.2	send_data_for_monitoring	102
5.3.4.3	use_ssl	102
5.3.4.4	organization	102
5.3.4.5	token	102
5.3.4.6	bucket	102
5.3.4.7	host	103
5.3.4.8	port	103
5.3.4.9	flushing_period	103
5.3.5	Группа параметров LUNA_SERVICE_METRICS	103
5.3.5.1	enabled	103
5.3.5.2	metrics_format	103
5.3.5.3	extra_labels	104
5.3.6	LUNA_STREAMS_HTTP_SETTINGS	104
5.3.6.1	request_timeout	104
5.3.6.2	response_timeout	104
5.3.6.3	request_max_size	104
5.3.6.4	keep_alive_timeout	104
5.3.7	LUNA_STREAMS_LOGS_CLEAR_INTERVAL	105
5.3.7.1	active	105

5.3.7.2	interval	105
5.3.7.3	interval_type	105
5.3.7.4	check_interval	105
5.3.8	Прочие	106
5.3.8.1	stream_worker_async_lock_timeout	106
5.3.8.2	stream_status_obsoleting_period	106
5.3.8.3	luna_streams_active_plugins	106
5.3.8.4	storage_time	106
6	Настройки FaceEngine	107
6.1	FaceDetV3	107
6.1.1	minFaceSize	107
6.2	HumanDetector	107
6.2.1	imageSize	107
7	Использование FaceStream с LUNA Configurator	108
7.1	Особенности работы с LUNA Configurator	108
7.2	Параметры в Configurator	108
7.3	Задание настроек для нескольких экземпляров FaceStream	109
8	Использование FaceStream с конфигурационными файлами	111
8.1	Ключи запуска с конфигурационными файлами	112
9	Пользовательский интерфейс сервиса LUNA Streams	114
9.1	Вкладка «Streams»	114
9.1.1	Создание потока	116
9.1.1.1	Группа «General stream parameters»	118
9.1.1.2	Группа «Stream data»	118
9.1.1.3	Группа «Stream handler parameters»	118
9.1.1.4	Группа «Geoposition»	119
9.1.1.5	Группа «Autorestart»	119
9.1.1.6	Группа «Sending parameters»	119
9.1.1.7	Группа «Use Primary Track»	120
9.1.1.8	Группа «Healthcheck parameters»	120
9.1.1.9	Группа «Liveness parameters»	121
9.1.1.10	Группа «Filtering parameters»	121
9.1.1.11	Группа «Additional parameters»	122
9.1.2	Редактирование потока	123
9.1.3	Удаление потока	123
9.2	Вкладка «Groups»	123
9.2.1	Привязка потока к группе	124

9.3	Вкладка «Queue»	125
10	Мониторинг	126
10.1	InfluxDB	126
10.2	Мониторинг FaceStream	127
10.2.1	Включение мониторинга	127
10.2.2	Отправляемые данные	127
10.3	Мониторинг LUNA Streams	129
10.3.1	Отправляемые данные в InfluxDB	129
10.3.2	Просмотр данных InfluxDB	131
10.3.3	Экспорт метрик в формате Prometheus	131
10.3.3.1	Тип метрик	131
10.3.4	Настройка сбора метрик для Prometheus	133
11	Вывод информации в логи	135
11.1	Формат вывода логов FaceStream	135
11.2	Ошибки сервиса LUNA Streams	135
11.2.1	Вернулся код ошибки 39001	136
11.2.2	Вернулся код ошибки 39002	136
11.2.3	Вернулся код ошибки 39003	136
11.2.4	Вернулся код ошибки 39004	137
11.2.5	Вернулся код ошибки 39005	137
11.2.6	Вернулся код ошибки 39006	137
11.2.7	Вернулся код ошибки 39007	138
11.2.8	Вернулся код ошибки 39008	138
11.2.9	Вернулся код ошибки 39009	138
11.2.10	Вернулся код ошибки 39010	139
12	Дополнительная информация	140
12.1	Диаграмма активности для Liveness и Primary Track Policy	140
12.2	Особенности работы с предварительным просмотром потока	141
12.3	Проксирование запросов в LUNA Streams через LUNA API	142
12.4	Данные событий, генерируемых с помощью FaceStream	146

Глоссарий

Термин	Значение термина
Батч	Группа данных, обрабатываемых одновременно.
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	Кадр видеопотока, на котором лицо/тело зафиксировано в оптимальном ракурсе для дальнейшей обработки.
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.
Трекинг	Функция отслеживания объекта (лица) на последовательности кадров.

Термин	Значение термина
LUNA Streams	Сервис для создания и управления потоками, которые содержат политики обработки видеопотока/видеофайла/набора изображений.

Аббревиатура	Расшифровка
БД, DB	База данных
LP	LUNA PLATFORM

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 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.

- **Проверка на Liveness**

Выполняется проверка на Liveness по одному или нескольким кадрам трека.

- **Отсылка лучших снимков 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 и тела. 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:

- только с лицами
- только с телами
- с лицами и телами (см. раздел [«Совместный режим детектирования лиц и тел»](#))

2.1 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».

2.1.1 Работа с лицами

Схема работы FaceStream с лицами представлена на рисунке ниже.

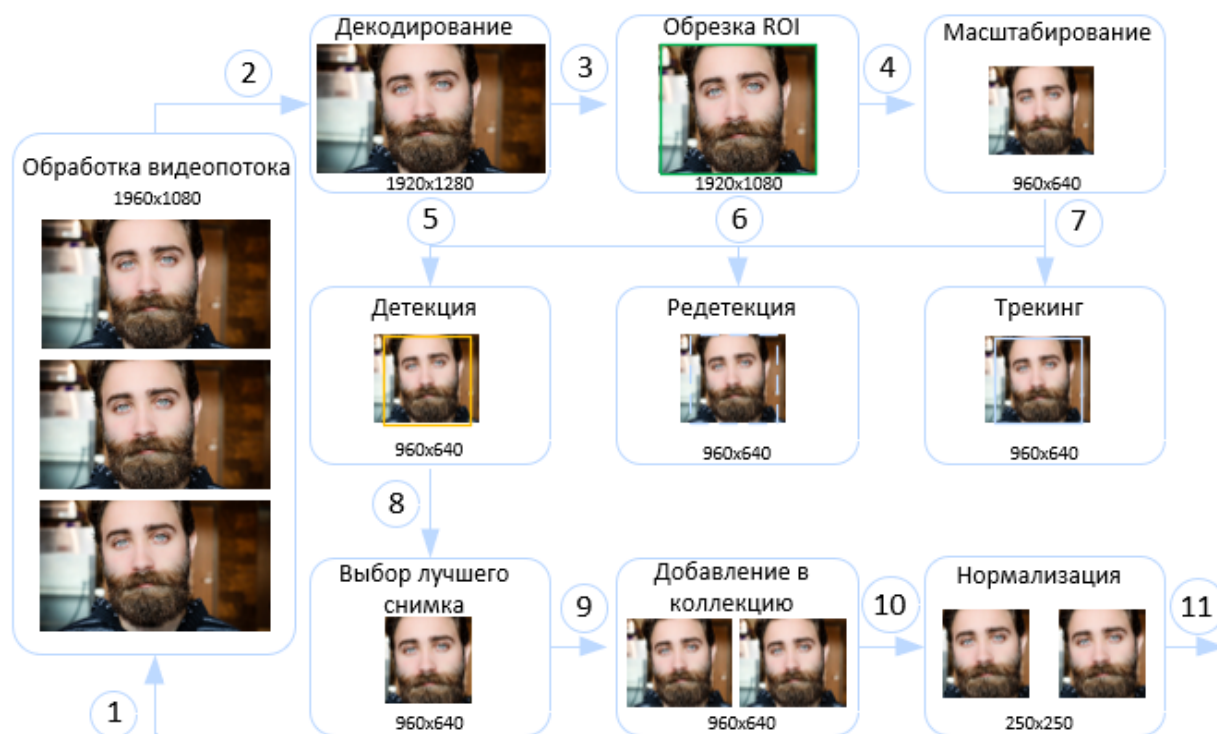


Рис. 1: Схема работы FaceStream с лицами

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 и дополнительные настройки обработки потока указываются в теле HTTP-запроса к сервису [LUNA Streams](#). Затем данные настройки извлекаются приложением FaceStream.
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. <http://git.visionlabs.ru/facestream/fs-extras/-/jobs/3952748/artifacts/browse>
8. FaceStream отсекает кадры плохого качества и выбирает лучшие снимки. Существует несколько алгоритмов выбора лучшей детекции/детекций в треке. См. описание группы параметров в разделе [«Фильтрация кадров»](#) (настройки управления потоками).
9. Если кадр является лучшим снимком, он добавляется в коллекцию лучших снимков. В зависимости от настройки [«face_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, исходные изображения (опционально) и дополнительная информация из [настроек управления потоком](#) отправляется в LUNA PLATFORM в виде HTTP-запроса к ресурсу `«/6/handlers/{handler_id}/stream_events»` для генерации события.

Основные параметры видеопотока (протокол передачи данных, путь к источнику, область интереса на кадре и др.) задается в разделе [«data»](#) (настройки управления потоками).

Адрес LUNA PLATFORM задается в разделе [«lunaplatform»](#) (настройки FaceStream).

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

2.1.2 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 (количество задается лицензией). Источники и дополнительные настройки обработки потока задаются с помощью отправки HTTP-запроса в сервис [LUNA Streams](#), которые в дальнейшем извлекаются приложением FaceStream.
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_body»](#) (настройки управления потоками).
8. Если кадр является лучшим снимком, он добавляется в коллекцию лучших снимков. В зависимости от параметра [«body_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.
10. Лучшие снимки, исходные изображения (опционально) и дополнительная информация из [настроек управления потоком](#) отправляется в LUNA PLATFORM в виде HTTP-запроса к ресурсу `«/6/handlers/{handler_id}/stream_events»` для генерации события. Вместе с лучшими снимками отправляются детекции с координатами тела человека. Количество отправляемых детекций задается в параметре [«minimal_body_track_length_to_send»](#) (настройки управления потоками).

Основные параметры видеопотока (протокол передачи данных, путь к источнику, область интереса на кадре и др.) задается в разделе [«data»](#) (настройки управления потоками).

Адрес LUNA PLATFORM задается в разделе [«lunaplatform»](#) (настройки FaceStream).

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

2.2 Взаимодействие FaceStream с LUNA Streams

Для работы FaceStream необходимо предварительно запускать дополнительный сервис — LUNA Streams (порт по умолчанию — 5160). В теле запроса [«create stream»](#) 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 OpenAPI LUNA Streams](#).

LUNA Streams имеет две версии API. В данной документации описывается вторая версия API. См. описание первой версии API в документации FaceStream v.5.1.49 и ниже.

LUNA Streams имеет собственный пользовательский интерфейс, предназначенный для работы с потоками. См. подробную информацию в разделе [«Пользовательский интерфейс сервиса 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.

2.2.1 Диаграмма последовательности создания и обработки потока

Диаграмма последовательности создания и обработки потока приведена ниже:

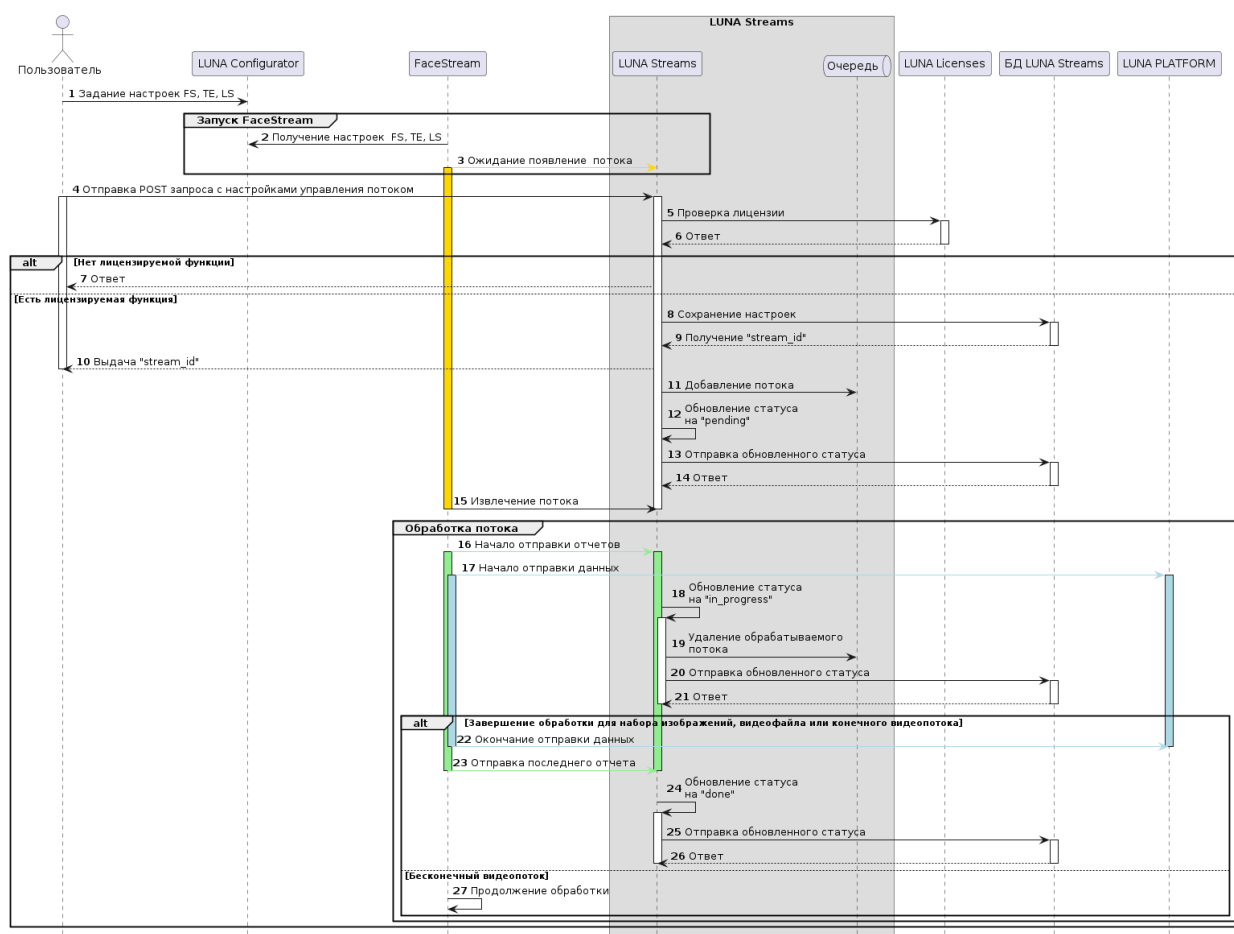


Рис. 3: Диаграмма последовательности создания и обработки потока

1☒ Пользователь задает настройки **FaceStream**, **TrackEngine** и настройки сервиса **LUNA Streams** в сервисе LUNA Configurator.

Настройки сервиса LUNA Streams содержат только настройки сервиса (адрес БД LUNA Streams, адрес InfluxDB и пр.) и не содержат настроек управления видеопотоком (параметры отправки лучших снимков, адрес LP, идентификаторы обработчиков и др.). Такие настройки задаются с помощью отдельного HTTP-запроса после запуска FaceStream (см. п.4).

При необходимости пользователь может задать настройки **FaceEngine** в отдельном конфигурационном файле.

2☒ При запуске FaceStream, приложение считывает настройки из сервиса LUNA Configurator.

3☒ FaceStream переходит в режим ожидания появления потока.

4☒ Пользователь отправляет HTTP-запрос «**create stream**» к сервису LUNA Streams, содержащий **настройки управления потоком** в теле запроса.

5❑ Сервис LUNA Streams проверяет наличие лицензируемой функции, регулирующей количество потоков для работы LUNA Streams, в сервисе LUNA Licenses. Если на данный момент уже обрабатываются какие-либо потоки, то дополнительно проверяется количество уже обрабатываемых потоков на момент запроса с помощью отчета FaceStream (не отражено на диаграмме, см. п. 17).

6❑ Сервис LUNA Licenses возвращает ответ.

7❑ Если лицензируемая функция отсутствует или на момент создания потока обрабатывается максимальное количество доступных потоков, то пользователю возвращается соответствующая ошибка.

8❑ Если лицензируемая функция присутствует и на момент создания потока еще не обрабатывается максимальное количество доступных потоков то сервис LUNA Streams создает поток и записывает настройки управления потоком в БД LUNA Streams.

9❑ Сервис LUNA Streams получает идентификатор «stream_id».

10❑ Сервис LUNA Streams возвращает пользователю уникальный идентификатор «stream_id».

11❑ Сервис LUNA Streams добавляет поток во внутреннюю очередь.

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

12❑ Сервис LUNA Streams обновляет статус потока на «pending».

13❑ Сервис LUNA Streams обновляет статус потока в БД LUNA Streams.

14❑ Сервис LUNA Streams получает ответ.

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.

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

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

15❑ FaceStream извлекает поток(и) из очереди со статусом «pending».

16❑ FaceStream начинает опрашивать данные в основные сервисы LUNA PLATFORM 5 для дальнейшей обработки кадров по заданному обработчику и генерации событий.

17❑ FaceStream начинает отправку отчетов об обработке потоков в LUNA Streams.

Время отправки отчетов является фиксированным и не может быть изменено.

18❑ Сервис LUNA Streams обновляет статус обрабатываемых потоков на «in_progress».

19❑ Сервис LUNA Streams удаляет обрабатываемый поток из очереди.

20❑ Сервис LUNA Streams обновляет статус потока в БД LUNA Streams.

21❑ Сервис LUNA Streams получает ответ.

Завершение обработки для набора изображений, видеофайла или конечного видеопотока:

22❑ FaceStream прекращает отправлять данные в LUNA PLATFORM.

Если необходимо, чтобы видеопоток воспринимался как конечный, то необходимо включить параметр [«endless»](#).

23❑ FaceStream отправляет последний отчет в сервис LUNA Streams.

Если в отчёте сказано, что какой-то поток был обработан, то FaceStream забирает следующие параметры потока со статусом «pending» из очереди LUNA Streams, 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.

24❑ Сервис LUNA Streams обновляет статус потока на «done» в БД Streams.

25❑ Сервис LUNA Streams обновляет статус потока в БД LUNA Streams.

26❑ Сервис LUNA Streams получает ответ.

Бесконечный видеопоток:

27❑ FaceStream будет отправлять данные в LUNA PLATFORM и отчеты в LUNA Streams до тех пор, пока видеопоток не будет прерван.

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

2.2.2 Запросы к сервису LUNA Streams

Доступны следующие запросы к сервису LUNA Streams для работы с потоками:

- задание [настроек для управления потоком](#) (запрос [«create stream»](#))
- получение существующих потоков по их «stream_id» с описанием данных каждого потока (запрос [«get streams»](#))
- получение всей информации по потоку по его «stream_id» (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 OpenAPI LUNA Streams](#).

2.2.3 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 «рабочего процесса»
- in_progress — поток обрабатывается
- done — поток полностью обработан (актуально для типов передачи потока [«videofile»/«images»](#) или [«tcp/udp»] с параметром [«endless»](#) равным «false»)
- pause — пользователь остановил обработку потока (не применимо для типов передачи потока [«videofile»](#) или [«images»](#))
- restart — сервер перезапустил обработку потока
- cancel — пользователь отменил обработку потока
- failure — при обработке потока произошла ошибка
- handler_lost — «рабочий процесс» FaceStream потерял, следует передать другому (не применимо для типов передачи потока [«videofile»](#) или [«images»](#))
- not_found — поток был в обработке, но в процессе выполнения был удалён
- deleted — намеренное удаление потока

Статусы «pause» и «cancel» можно указать при обновлении потока с помощью запроса [«update stream»](#).

Статусы «restart», «handler_lost» являются временными. При данных статусах невозможно получить поток, однако переход через эти статусы регистрируется в обычном режиме. Статус «restart» может появиться только при использовании группы настроек «autorestart» (см. раздел [«Автоматический перезапуск потоков»](#) ниже).

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.

2.2.3.1 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_-
	Нет	pending	progress	done	restart	pause	cancel	failure	lost
Нет		+				+			
pending	+		+		+	+	+		
in_progress	+	+		+	+	+	+	+	+
done	+	+			+	+			
restart		+				+			
pause	+	+			+		+		
cancel	+	+			+	+			
failure	+	+			+	+			
handler_- lost					+				

* не поддерживается для типов передачи потока «[videofile](#)» или «[images](#)»

2.2.4 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 «рабочий процесс» свободного потока с запросом на пул из очереди, поток принимается к обработке и ему присваивается статус «in_progress».

После того, как поток был обработан «рабочим процессом» FaceStream, ему присваивается статус «done» в случае успеха (актуально для типов передачи потока «[videofile](#)»/«[images](#)» или «[tcp/udp](#)» с параметром «[endless](#)» равным «false») или «failure», если произошли какие-либо ошибки. Однако статус обработки потока может быть понижен с «in_progress» по следующим причинам:

- отсутствие обратной связи от «рабочего процесса» 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

Для типов передачи потока «[tcp](#)» или «[udp](#)» с параметром «[endless](#)» равным «true», статус не может измениться на «done».

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

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».

Потоки со статусом «failure» могут быть автоматически перезапущены.

2.2.4.1 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.

Ниже описан процесс обработки потоков с включенным автоматическим перезапуском.

Когда происходит попытка автоматической перезагрузки потока данных, происходят следующие изменения:

- **статус потока** изменяется сначала на «restart», а затем на «pending»;
- счетчик попыток автоматической перезагрузки «current_attempt» увеличивается на 1;
- запись времени последней попытки «last_attempt_time» обновляется, чтобы отразить актуальное время.

Для того чтобы произошла перезагрузка, должны соблюдаться следующие условия:

- статус потока должен находиться в состоянии «failure»;
- должен быть включен автоматический перезапуск потока (параметр «restart»);
- значение текущей попытки автоматической перезагрузки «current_attempt» должно быть равно «null» или меньше, чем максимальное число попыток «attempt_count»;

- время последней попытки автоматической перезагрузки «last_attempt_time» должно быть равно «null» или разница между текущим временем и временем последней попытки должна быть больше или равна задержке «delay».

Если соблюдаются условия ниже, то произойдет сбой автоматического перезапуска потока (прекращение попыток перезапуска):

- статус потока находится в состоянии «failure»;
- статус автоматического перезапуска потока находится в состоянии «in_progress»;
- значение текущей попытки автоматической перезагрузки «current_attempt» равно значению максимального числа попыток «attempt_count».

Если соблюдаются условия ниже, то произойдет завершение работы автоматического перезапуска потока:

- статус потока не равен статусу «failure»;
- статус автоматической перезагрузки потока находится в состоянии «in_progress»;
- время последней попытки автоматической перезагрузки «last_attempt_time» равно «null» или разница между текущим временем и временем последней попытки больше или равна задержке «delay».

Завершение работы автоматического перезапуска потока означает смену статуса автоматической перезагрузки потока на «enabled» и сброс значений текущей попытки автоматической перезагрузки «current_attempt» и времени последней попытки автоматической перезагрузки «last_attempt_time».

2.2.5 Автоматическое удаление логов потоков

При необходимости можно запустить автоматическое удаление логов потоков. Автоматическое удаление логов помогает очистить таблицу «log» базы данных LUNA Streams от большого количества ненужных логов.

Самая последняя запись для каждого потока удалена не будет.

Автоматическое удаление логов потоков настраивается с помощью нижеописанных параметров из группы параметров «LUNA_STREAMS_LOGS_CLEAR_INTERVAL»:

- «interval». Задаёт интервал удаления логов. Логи, старше чем данный интервал, будут удалены.
- «interval_type». Задаёт тип интервала (недели, дни, часы, минуты, секунды).
- «check_interval». Задаёт частоту проверки логов на удаление (секунды).
- «active». Включает/выключает автоматическое удаление логов потоков.

По умолчанию автоматическое удаление логов выключено («active» = false).

Значения настроек по умолчанию включают автоматическое удаление логов («active» = true) с проверкой потоков логов в базе данных каждые 180 секунд («check_interval» = 180) и удаляют логи старше 7 дней («interval» = 7 и «interval_type» = days).

Пример проверки логов на удаление каждые 5 минут и удаления логов старше 4 недель:

```
{
  "interval": 4,
  "interval_type": "weeks",
  "check_interval": 300,
  "active": true
}
```

2.2.6 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».

2.2.7 Описание базы данных LUNA Streams

Общая схема базы данных LUNA Streams приведена ниже.

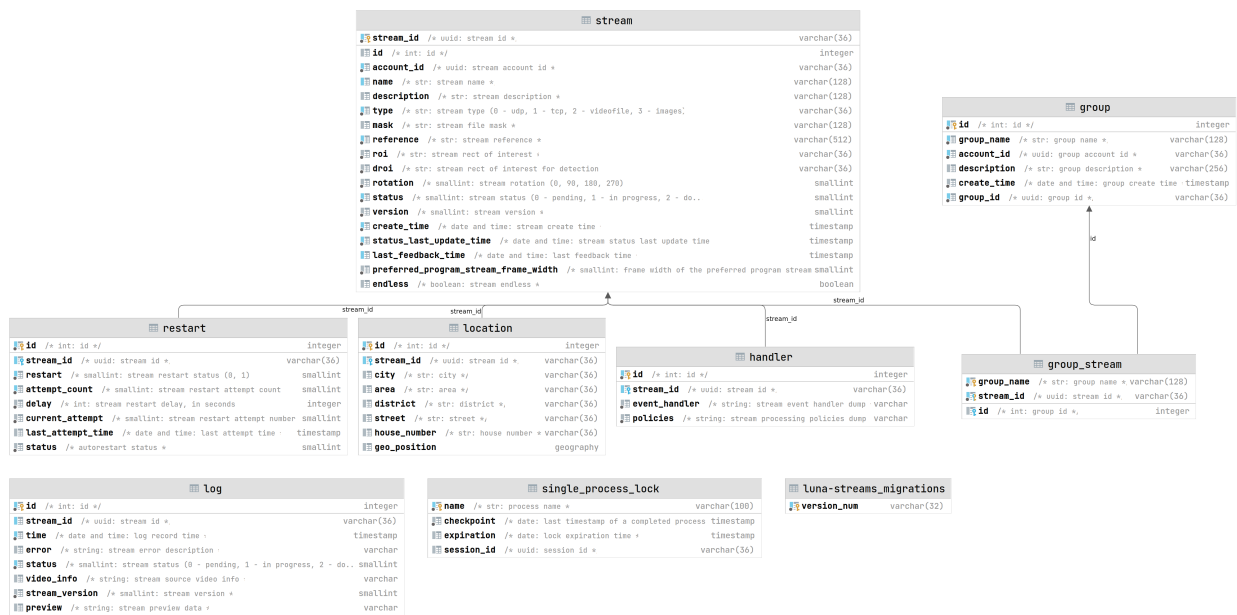


Рис. 4: База данных сервиса LUNA Streams

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

2.3 Рекомендации по настройке FaceStream

Данный раздел содержит общие рекомендации по настройке FaceStream.

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

2.3.1 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»`.

2.3.2 Настройка производительности FaceStream

Описанные ниже настройки оказывают наибольшее влияние на производительность FaceStream.

2.3.2.1 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.

2.3.2.2 Масштабирование кадра

Масштабирование кадра перед началом обработки позволяет значительно увеличить производительность FaceStream. Масштабирование кадра можно включается в следующих настройках:

Режим работы	Название параметра	Значение по умолчанию	Состояние по умолчанию
Только с детектором лиц	«minFaceSize» настроек FaceEngine	50 (уменьшить в 2.5 раза)	Включено
Только с детектором тел	«imageSize» настроек FaceEngine	640 (уменьшить до 640 пикс. по наибольшей стороне)	Включено
С детектором лиц и с детектором тел	«detector-scaling» и «scale-result-size» настроек TrackEngine	640 (уменьшить до 640 пикс. по наибольшей стороне)	Включено

2.3.3 Определение области с движением

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

- [frg-subtractor](#) — включение режима 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

2.3.4 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.

2.3.5 Общая информация о настройке

2.3.5.1 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.

2.3.5.2 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 `«face_bestshots_to_send»` или `«body_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»`.

Примечание. Параметр `«silent_period»` работает только при включенном режиме работы с лицами. При работе с телами или в [совместном режиме](#) параметр не будет работать.

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

- 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
```

2.3.5.3 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_face»);
- 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 «face_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_body».

2.3.5.4 Работа со СКУД

Работа со СКУД ведётся 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 потребуется обновить 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.

2.4 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.

2.4.1 Форматы видео

Основные форматы видео, обрабатываемые FaceStream:

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

2.4.2 Стандарты сжатия видео (кодеки)

Основные стандарты сжатия видео, с которыми работает FaceStream:

- MPEG1, MPEG2, MPEG3, MPEG4,
- MS MPEG4,
- MS MPEG4v2,
- MJPEG,
- H.264,
- H.265,
- VC1,
- HEVC,
- VP8,
- VP9,
- AV1,
- другие.

2.4.3 Протоколы

Основные протоколы транспортного уровня, используемые FaceStream для получения данных:

- TCP,
- UDP.

Основные протоколы прикладного уровня, используемые FaceStream для получения данных:

- HTTP (основывается на протоколе транспортного уровня **TCP**)
- HTTPS (основывается на протоколе транспортного уровня **TCP**)
- RTP (основывается на протоколе транспортного уровня **UDP**)
- RTSP (основывается на протоколах транспортного уровня **TCP** или **UDP**)
- RTMP (основывается на протоколе транспортного уровня **TCP**)
- HLS (основывается на протоколе транспортного уровня **TCP**)

Для использования протоколов прикладного уровня, необходимо указать соответствующий протокол транспортного уровня в параметре «type» настроек управления потоками.

2.5 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).

2.6 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 значения, заданные в параметрах «[min_score_face](#)» или «[min_score_body](#)».

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

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

Ограничивающий прямоугольник **оранжевого цвета** возникает при использовании ROI.

2.7 Совместный режим детектирования лиц и тел

В FaceStream доступен режим одновременного детектирования лиц и тел.

Важно! Функционал находится в состоянии бета-тестирования. Часть функций может не работать.

Совместный режим включается с помощью одновременного включения настроек «[use-face-detector](#)» и «[use-body-detector](#)». Использование совместного режима регулируется настройкой «[data](#)» > «[analytics](#)» > «[mode](#)» в настройках создания потока.

При включении совместного режима, FaceStream отправляет следующие данные к ресурсу `handlers/{handler_id}/stream_events` LUNA PLATFORM:

- Лучшие снимки лиц и тел
- Время детекции
- Время относительно начала видеофайла
- Координаты лиц и тел
- Исходные изображения

В совместном режиме используется порог «min_score_body».

Периодическая отправка лучших снимков в совместном режиме работает аналогично работе с телами. Это значит, что параметр «silent_period» не будет работать.

2.7.1 Неподдерживаемый функционал

В настоящее время в совместном режиме не поддерживается следующий функционал:

- Политика Главного трека.
- Параметр «data» > «analytics» > «sending» > «full_frame_settings», определяющий какие исходные кадры будут отправлены (лица, тела или лица и тела). В настоящий момент отправляются только исходные кадры лиц и тел.
- Периодическая отправка лиц. В настоящий момент работает по сценарию периодической отправки тел.
- Параметр «data» > «analytics» > «sending» > «bestshot_settings» > «type», определяющий какие лучшие снимки будут отправлены (лица, тела или лица и тела). В настоящий момент будут отправляться только лучшие снимки лиц и тел.

3 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 6: 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 7: Настройки управления потоками

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

3.1 Настройки для работы с лицами

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

Таблица 8: Настройки TrackEngine

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

Таблица 9: 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>lunaplatform > api_version</code>	Позволяет задать версию API сервиса LUNA API для отправки запросов в LP — «6»
<code>lunaplatform > origin</code>	Полный сетевой путь до LUNA PLATFORM
<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 10: 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
data > analytics > event_handler > handler_id	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.

3.2 Настройки для работы с телами

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

Таблица 11: Настройки TrackEngine

Параметр	Описание
use-face-detector	Позволяет обнаруживать лица — «0»
use-body-detector	Позволяет обнаруживать тела — «1»

Таблица 12: Настройки FaceStream

Параметр	Описание
lunastreams > api_version	Позволяет задать версию API сервиса LUNA Streams для отправки запросов в LP — «1»
lunastreams > origin	Полный сетевой путь до сервиса LUNA Streams
lunaplatform > api_version	Позволяет задать версию API сервиса LUNA API для отправки запросов в LP — «6»
lunaplatform > origin	Полный сетевой путь до LUNA PLATFORM
sending > aggregate_attr_requests	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 13: 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>data > analytics > event_handler > handler_id</code>	обработчик LP, позволяющий гибко настроить обработку тел — «строка в формате UUID»

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

4 Настройки управления потоками

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

Важно! LUNA Streams имеет две версии API. В данной документации описывается вторая версия API. См. описание первой версии API в документации FaceStream v.5.1.49 и ниже.

Важно! Настройки управления потоками не хранятся в сервисе LUNA Configurator и могут быть заданы только с помощью HTTP запросов к сервису LUNA Streams. Настройки сервиса LUNA Streams, задаваемые в LUNA Configurator, описаны в разделе «[Настройки LUNA Streams](#)».

Базовое описание параметров приведено в [спецификации OpenAPI сервиса LUNA Streams](#). В данном разделе приводится расширенное описание для некоторых параметров.

Фактически, все настройки управления потоками можно разделить на следующие категории:

- [общие настройки потока](#) — имя потока, описание, политика перезапуска при ошибке и пр.
- [настройки обработки потока](#) — информация о типе потока, его адресе и некоторую дополнительную информацию
- [настройки аналитики](#) — анализ содержимого потока и настройка фильтрации для отправки во внешнюю систему
- [настройки Liveness](#)
- [настройки политики Главного трека](#)

4.1 Общие настройки потока

Общие настройки позволяют определить ключевые параметры, необходимые для авторизации, управления и характеристики потока.

Общие настройки потока:

- «account_id» — задает значение обязательного поля «account_id», которое передаётся в заголовке запроса в LUNA PLATFORM 5 в сервис LUNA API. Параметр используется для привязки получаемых данных к конкретному пользователю.
- «name» — имя потока. Служит для идентификации источника отправляемых кадров.
- «description» — пользовательское описание потока
- «location» — задает информацию о расположении источника видеосигнала (город, область, район и т.д.)
- «autorestart» — настройка автоматического перезапуска потока. См. раздел [«Автоматический перезапуск потоков»](#) для более подробной информации.
- «status» — статус потока на момент начала обработки
- «group_name» и «group_id» — параметры привязки потока к [группе](#)

Также доступна группа параметров «data», задающая основные настройки для обработки потока. См. раздел [«Настройки обработки потока»](#) для более подробной информации.

4.2 Настройки обработки потока

Для начала обработки FaceStream должен получить информацию о типе потока, его адресе и некоторую дополнительную информацию. Соответствующие параметры задаются в группе параметров «data».

Основные настройки в группе параметров «data»:

- «type» — тип передачи потока:
 - сетевой протокол TCP
 - сетевой протокол UDP
 - набор изображений
 - видеофайл
- «reference» — источник потока (ссылка, путь к видеофайлу или набору изображений и др.)
- «roi» — область интереса в которой происходит детекция и сопровождение лица на кадре
- «ffmpeg_threads_number» — количество потоков для декодирования видео с помощью FFmpeg
- «real_time_mode_fps» — количество FPS, с которым будет обработан видеофайл
- «frame_processing_mode» — параметр, определяющий будет ли обрабатываться полный или масштабированный кадр
- «rotation» — угол поворота изображения с источника. Используется в случае, если входящий поток повернут, например, если камера установлена на потолке. Поворот выполняется по часовой стрелке.
- «preferred_program_stream_frame_width» — позволяет автоматически выбирать оптимальный канал из нескольких в потоке
- «endless» — управление перезагрузкой потока при получении ошибки сети
- «mask» — маска имён файлов в директории с изображениями

Также в группе параметров «data» задаются настройки аналитики. См. раздел «[Настройки аналитики](#)» для более подробной информации.

4.2.1 type

Тип передачи потока. После выбора типа передачи потока, необходимо указать путь к источнику/изображениям/USB-устройству и тд. в настройке [«reference»](#).

Приложение может использовать один из следующих типов передачи потока:

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

В FaceStream указываются только протоколы транспортного уровня (TCP или UDP). Необходимо понимать на каком протоколе транспортного уровня базируется протокол прикладного уровня (HTTP, RTSP, HLS и др.). См. подробную информацию в разделе [«Протоколы»](#).

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.

FaceStream обрабатывает данные из типов **images** и **videofile** только **один** раз. После обработки всех изображений или видеофайла в логи FaceStream будет выведено сообщение о завершении обработки. Если набор изображений или видеофайлы были изменены, то необходимо перезапустить обработку потока, после чего FaceStream снова единоразово выполнит обработку всех изображений или видеофайла.

4.2.2 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": "/example/path/to/video/video.mp4"
```

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

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

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

4.2.3 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 параметрами «x», «y», «width», «height» и «mode», где:

- «x» и «y» – координаты верхней левой точки области интереса ROI;
- «width» и «height» – ширина и высота обрабатываемой области кадра;
- «mode» – режим указания «x», «y», «width» и «height». Доступно два режима:
 - «abs» – параметры «x», «y», «width» и «height» задаются в пикселях;
 - «percent» – параметры «x», «y», «width» и «height» задаются в процентах от текущего размера кадра.

Если поле «mode» не указывается в теле запроса, то будет использовано значение «abs».

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

Система координат на изображении задается аналогично рисунку ниже.

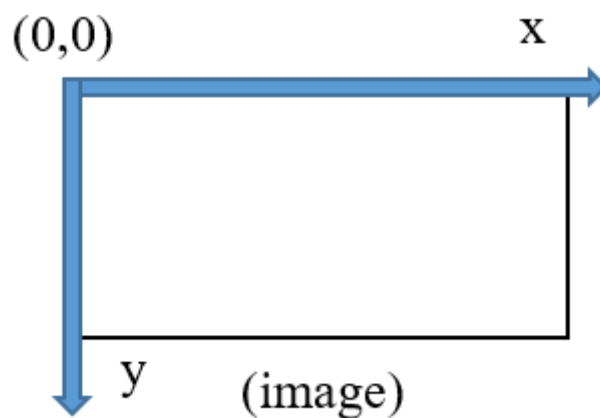


Рис. 7: Система координат ROI

Ниже приведен пример расчета ROI в процентах:

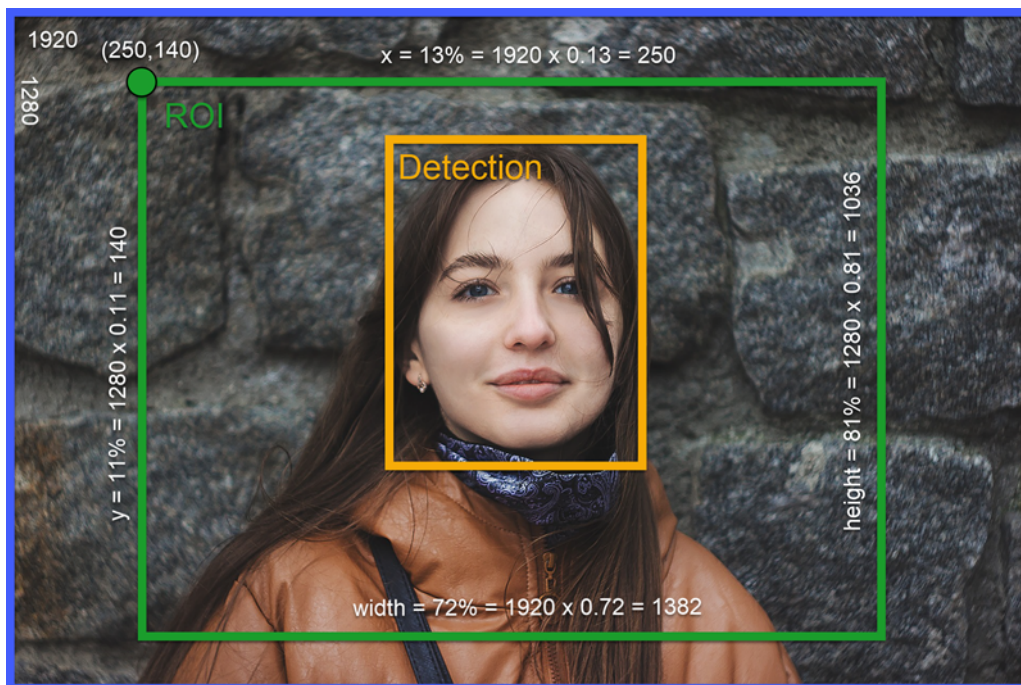


Рис. 8: Пример расчета ROI в процентах

4.2.4 frame_processing_mode

Данный параметр используется 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 TrackEngine (стандартное поведение для релизов 3.2.4 и ниже).

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

4.2.5 real_time_mode_fps

Данный параметр используется только для типа «videofile».

Параметр позволяет задать количество FPS, с которым будет обработан видеофайл.

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».

4.2.6 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).

4.2.7 preferred_program_stream_frame_width

Данный параметр используется для типов «tcp», «udp».

Данный параметр 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. Если указывается значение «0», то будет использоваться канал по умолчанию. Можно задать значение очень большим, чтобы заведомо выбрать наибольший канал.

В логах FaceStream (тег «ffmpeg») выбранное значение будет указываться в виде сообщения:

```
Url: [url] selected preferred stream #[res], [width]x[height] preferred  
width: [preferredProgramStreamFrameWidth]
```

Значение по умолчанию равно 800.

4.2.8 endless

Данный параметр используется для типов «tcp», «udp».

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

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.

4.2.9 mask

Данный параметр используется для типа «images».

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-000000001.jpg
- Photo-000000002.jpg
- Photo-000000003.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"
```

4.3 Настройки аналитики

Под аналитикой понимается анализ содержимого видеопотока для извлечения ключевых данных и характеристик. Задание настроек аналитики также предоставляет возможность настройки фильтрации данных и их отправки во внешнюю систему. Соответствующие параметры задаются в группе параметров «analytics».

Основные настройки в группе параметров «analytics»:

- «enabled» — включение аналитики
- «mode» — режим аналитики (1 — только лица, 2 — только тела, 3 — [тела и лица](#))
- «droi» — область выбора исходного кадра
- «filtering» — объекты фильтрации изображений и отправки результирующих лучших кадров
- «sending» — настройка параметров, связанных с составлением коллекции лучших кадров, а также периода, в течении которого будет проводиться анализ кадров для выбора лучшего кадра
- «event_handler» — настройка параметров, связанных с интеграцией с внешней системой для последующей обработки кадров
- «healthcheck» — настройка параметров, отвечающих за повторное подключение к потоку при возникновении ошибок во время потоковой передачи видео

Также в группе параметров «analytics» задаются настройки проверки Liveness и политики Главного трека. Как правило, данные параметры редко используются в кооперативном режиме. См. разделы [«Настройка Liveness»](#) и [«Настройка политики Главного трека»](#) для более подробной информации

4.3.1 Область выбора лучшего кадра

Данный параметр используется только для работы с лицами.

Область интереса внутри исходного кадра или ROI задается с помощью параметра «droi». Если используется область ROI, то 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. Ни одна сторона детекции не должна выступать из области DROI даже на миллиметр.

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 параметрами «x», «y», «width», «height» и «mode», где:

- «x» и «y» – координаты верхней левой точки области интереса DROI;
- «width» и «height» – ширина и высота обрабатываемой области кадра;
- «mode» – режим указания «x», «y», «width» и «height». Доступно два режима:
 - «abs» – параметры «x», «y», «width» и «height» задаются в пикселях;
 - «percent» – параметры «x», «y», «width» и «height» задаются в процентах от текущего размера кадра.

Если поле «mode» не указывается в теле запроса, то будет использовано значение «abs».

При расчете DROI нужно учитывать то, что данная область интереса рассчитывается относительно исходного кадра, а не относительно ROI.

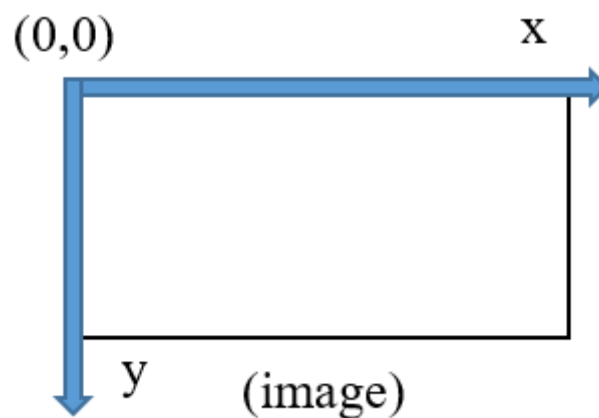


Рис. 9: Система координат DROI

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.

Ниже приведен пример расчета DROI в процентах:

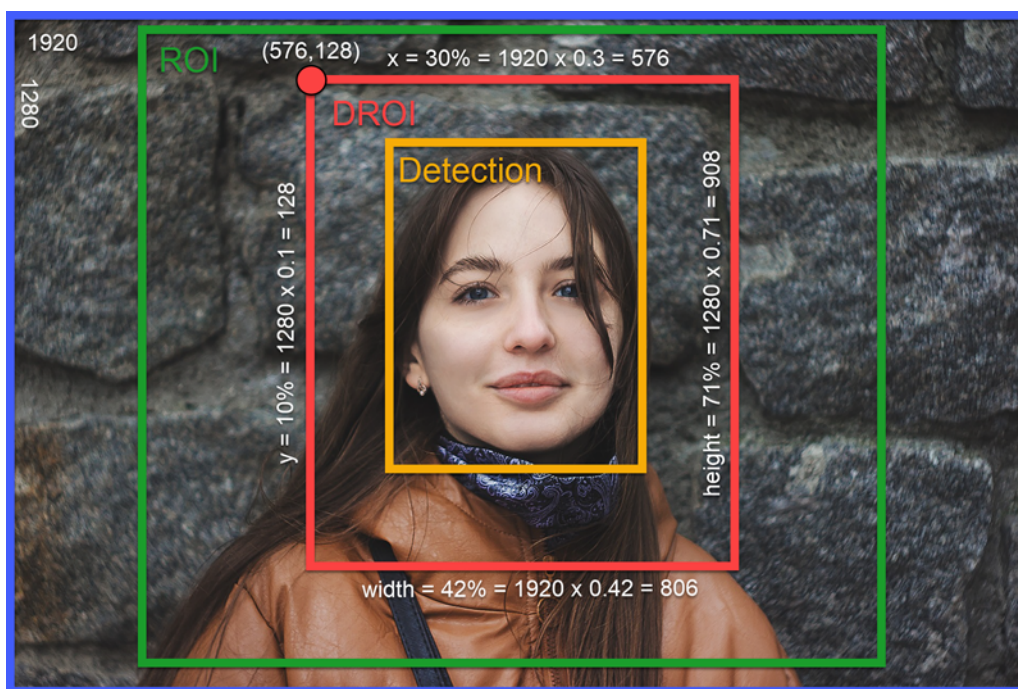


Рис. 10: Пример расчета DROI в процентах

4.3.2 Фильтрация кадров

Группа параметров «filtering» описывает объекты фильтрации изображений и отправки результатов лучших кадров.

4.3.2.1 min_score_face

Параметр задает порог для фильтрации отправляемых детекций лиц.

Для каждой детекции вычисляется AGS (Approximate Garbage Score) и общее качество.

Детекции, у которых значение AGS меньше, чем значение, указанное в порог «min_score_face» не будут считаться приемлимыми для дальнейшей работы. Далее по отфильтрованным детекциям будут выбраны лучшие снимки в соответствии с общим качеством детекции.

Если в параметре «min_score_face» задано значение «0», то лучшие снимки будут определены по общему качеству детекций.

Значение по умолчанию – **0.5187**.

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

4.3.2.2 min_score_body

Параметр задает порог для фильтрации отправляемых детекций тел.

Для каждой детекции вычисляется общее качество.

Детекции, у которых общее качество меньше, чем значение, указанное в порог «min_score» не будут считаться приемлимыми для дальнейшей работы. Далее по отфильтрованным детекциям будут выбраны лучшие снимки.

Если в параметре «min_score» задано значение «0», то лучшие снимки будут определены по качеству детекций.

Значение по умолчанию – **0.5**.

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

4.3.2.3 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.

Для отключения фильтрации необходимо задать значение «180».

4.3.2.4 `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.

Для отключения фильтрации необходимо задать значение «180».

4.3.2.5 `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.

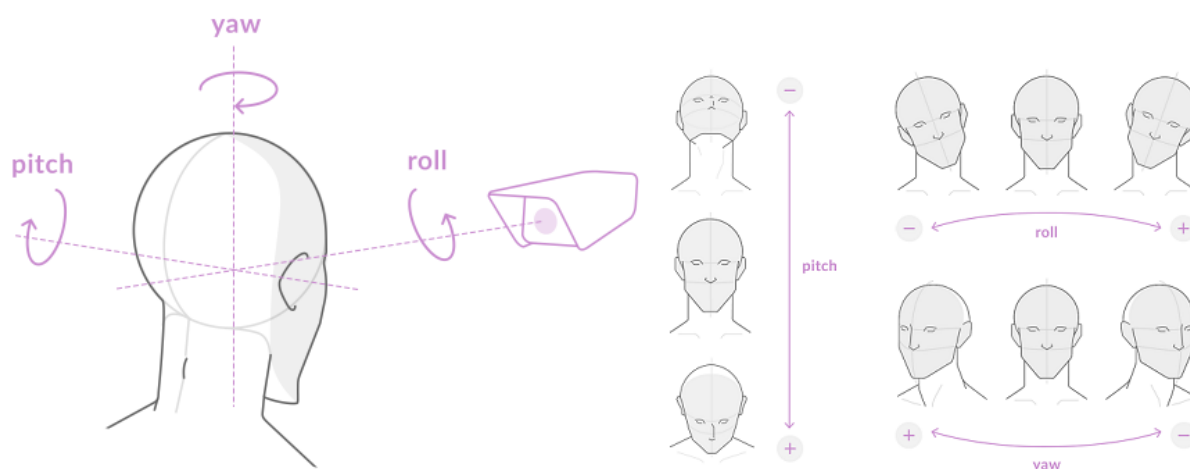


Рис. 11: Иллюстрация положения головы

Для отключения фильтрации необходимо задать значение «180».

4.3.2.6 `yaw_number`

Данный параметр используется только для работы с лицами.

Параметр определяет количество кадров, используемых для фильтрации фотоизображений по углу поворота (yaw angle) головы. Фильтрация отсекает изображения с сильно повернутыми от камеры лицами.

Как это работает:

В параметре задаётся количество кадров. Специальный алгоритм анализирует угол 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».

4.3.2.7 yaw_collection_mode

Данный параметр используется только для работы с лицами.

Параметр указывает системе, что необходимо собрать заданное в параметре «yaw_number» количество кадров для анализа угла поворота (yaw angle) головы.

Если параметр «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.

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

4.3.2.8 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. Филтрация проводится при значении параметра «0.3» и выше. При значении ниже филтрация **отключена**.

4.3.2.9 min_body_size_threshold

Параметр задает размер детекции тела, меньше которого она не будет рассматриваться как лучший снимок. Размер высчитывается как квадратный корень из произведения ширины (в пикселях) на высоту (в пикселях) детекции.

Пример: `min_body_size_threshold = sqrt (64*128)= 90.5`

Если значение равно «0», то фильтрация детекции тела по размеру выполняться не будет.

4.3.3 Отправка во внешнюю систему

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

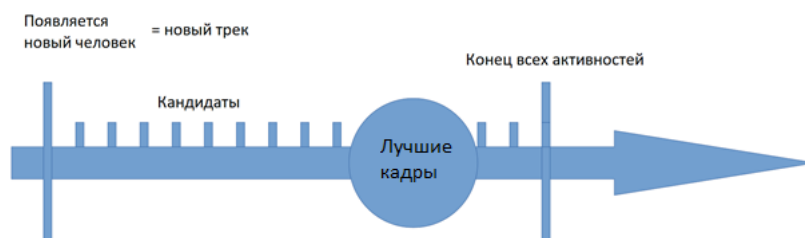


Рис. 12: Лучшие кадры

4.3.3.1 Настройки отправки данных тел

Вместе с исходными кадрами тел можно отправлять и детекции с координатами тела человека для того, чтобы можно было отследить путь человека. Настройки отправки данных тел регулируются в группе параметров «body».

Доступно два параметра:

- «delete_track_after_sending» — включить удаление лучших кадров и детекций после отправки данных. Если значение равно «false» (по умолчанию), то данные останутся в памяти.
- «send_only_full_set» — включить отправку данных во внешнюю систему только если набрано определенное количество лучших снимков (параметр «body_bestshots_to_send») и только если набрано определенное количество детекций с координатами тела человека (параметр «minimal_body_track_length_to_send» настроек FaceStream).

4.3.3.2 Настройки отправки лучших снимков

Настройки отправки лучших снимков регулируются в группе параметров «bestshot_settings».

Доступны следующие параметры:

- «type» — режим отправки лучших снимков:
 - только лучшие снимки лиц
 - только лучшие снимки тел
 - лучшие снимки лиц и тел в одном запросе
 - лучшие снимки лиц и тел в разных запросах
- «face_bestshots_to_send» и «body_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 во внешнюю систему.

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

4.3.3.3 full_frame_settings

Важно! В настоящий момент в [совместном режиме детектирования лиц и тел](#) отправляются только исходные кадры лиц и тел.

Данный параметр задает режим отправки исходных кадров:

- только исходные кадры лиц
- только исходные кадры тел
- исходные кадры лиц и тел в одном запросе
- исходные кадры лиц и тел в разных запросах

Важно! Отправка исходных кадров должна быть включена в параметре «[send_source_frame](#)» в настройках FaceStream.

4.3.3.4 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.

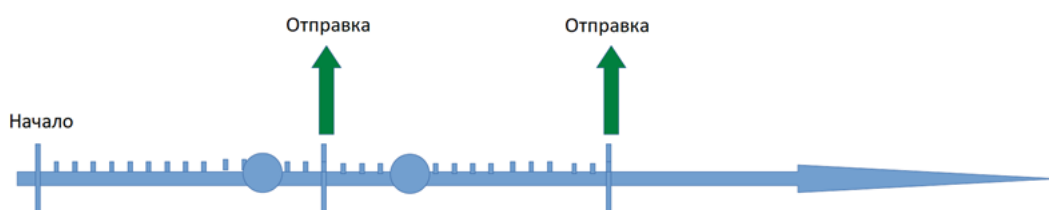


Рис. 13: Период отправки

Тип измерения задается в параметре «type» (см. ниже). Если значение равно «-1» (по умолчанию), то анализ кадров проводится по всем кадрам до конца трека. По окончании трека (когда объект покидает пределы кадра) лучший кадр будет отправлен во внешний сервис.

4.3.3.5 silent_period

Данный параметр используется только для работы с лицами.

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



Рис. 14: Период ожидания

Тип измерения задается в параметре «type» (см. ниже). Если значение равно «-1», то период ожидания будет длиться бесконечно.

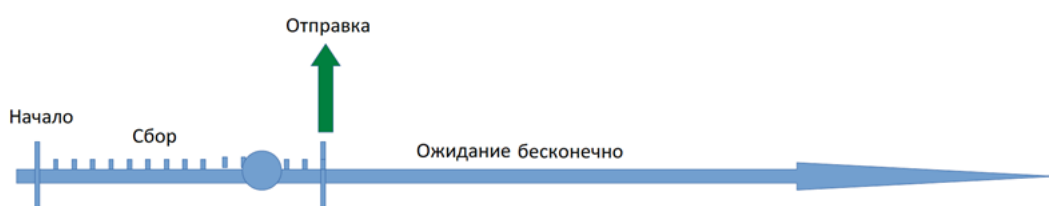


Рис. 15: Бесконечный период ожидания

4.3.3.6 type

Параметр задаёт тип измерения для параметров «silent_period» и «time_period_of_searching». Может принимать два значения - «frames» (кадры) или «sec» (секунды).

4.3.4 Настройки интеграции с внешней системой

Отправляемые FaceStream кадры могут обрабатываться внешней системой. Для этого требуется указать:

- идентификатор обработчика (параметр «handler_id»)
- [адрес для сохранения исходных кадров](#)
- [настройки авторизации через токен](#)

4.3.4.1 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"
```

Здесь:

- 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"
```

Здесь:

- 127.0.0.1 — IP адрес, где развернут сервис LUNA API;
- 6 — 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.

4.3.4.2 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. account_id).

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

4.3.5 Настройки healthcheck

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.

4.3.5.1 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.

4.3.5.2 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»](#).

4.3.5.3 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»](#).

4.3.5.4 timeout

Задаёт таймаут в миллисекундах на чтение закодированного пакета.

Благодаря этому параметру можно обеспечить более гибкую обработку видеопотока, контролировать скорость чтения видеопакетов и предотвращать переполнение буфера данных.

4.4 Настройки Liveness

Liveness не используется для типа «images».

Liveness используется только при работе с лицами.

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

Настройки Liveness задаются в группе параметров «data» > «analytics» > «liveness».

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

4.4.1 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.

4.4.1.1 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.

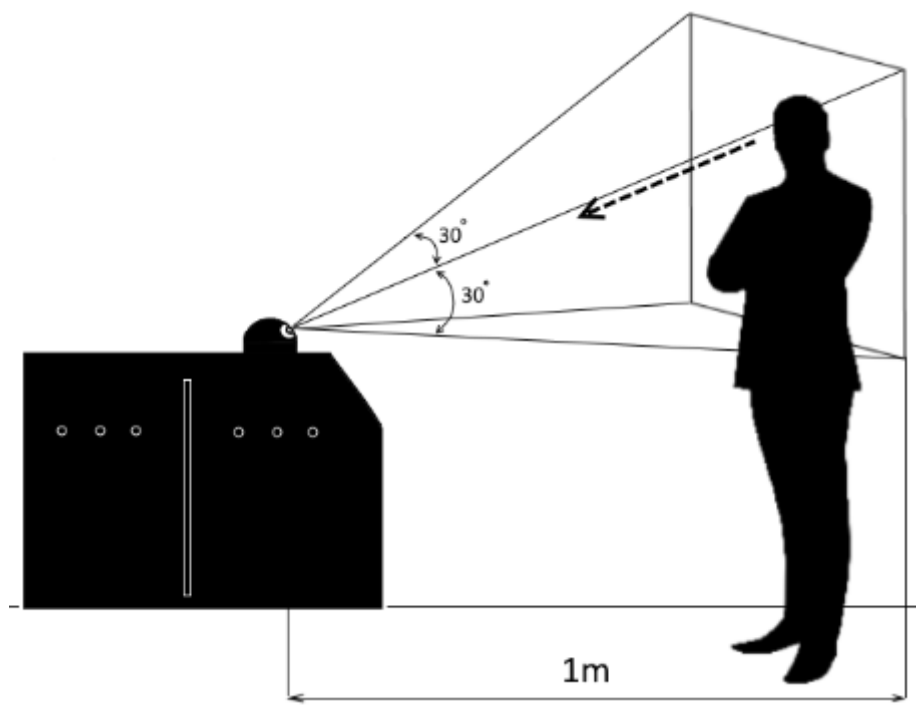


Рис. 16: Правильное расположение камеры для использования Liveness

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.

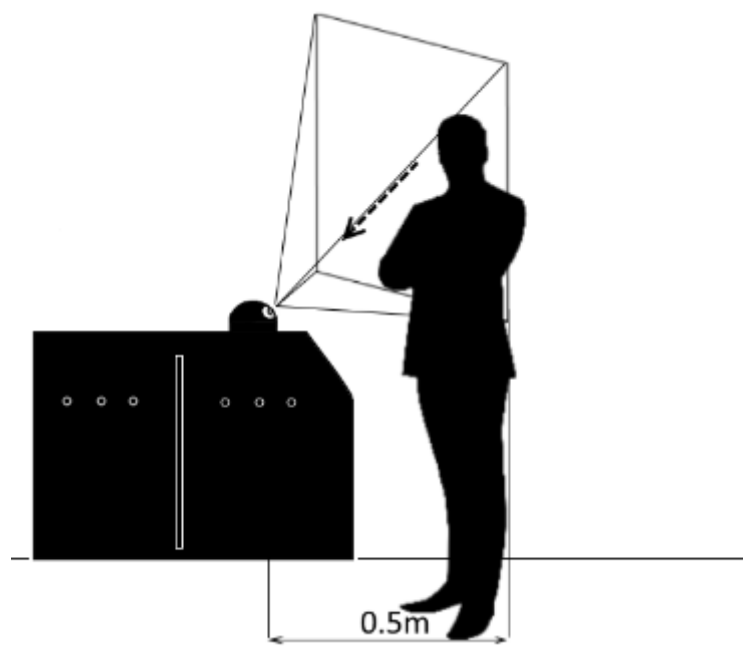


Рис. 17: Неправильное расположение камеры для использования Liveness

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.

4.4.1.2 Рекомендации по настройке FS

Рекомендуемые значения параметров секции «Liveness» приведены ниже.

```
"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.

4.4.1.3 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.

Часто вместе с Liveness используется политика Главного Трека. См. [диаграмму активности FaceStream при использовании Liveness и Primary Track Policy](#).

4.4.2 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).

4.4.3 use_flying_faces_liveness_filtration

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

4.4.4 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`.

4.4.5 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.

4.4.6 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».

4.4.7 livenesses_weights

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

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

- «use_mask_liveness_filtration»
- «use_flying_faces_liveness_filtration»

Величины указываются в долях единицы. В приведенном ниже примере 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": [0.25, 0.75]
```

4.4.8 mask_backgrounds_count

Количество кадров фона, которые используются для соответствующей проверки.

Изменение данного параметра запрещено.

4.5 Настройки политики Главного трека

Политика Главного трека используется только для работы с лицами.

Политика Главного трека не используется для типа «images».

Важно! В настоящий момент политика Главного трека не работает для [совместного режима детектирования лиц и тел](#).

Политика Главного трека 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.

Настройки политики Главного трека задаются в группе параметров «data» > «analytics» > «primary_track_policy».

Часто вместе с политикой Главного Трека используется Liveness. См. [диаграмму активности FaceStream при использовании Liveness и Primary Track Policy](#).

4.5.1 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 в LUNA PLATFORM.

Все остальные треки обрабатываются в обычном режиме, но отправка лучшего кадра в LUNA PLATFORM 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).

4.5.2 best_shot_min_size

Параметр используется при включенном параметре [«use_primary_track_policy»](#).

Задаёт минимальную высоту детекции в пикселях, при котором начинается анализ кадров и определение лучшего кадра.

4.5.3 **best_shot_proper_size**

Параметр используется при включенном параметре [«use_primary_track_policy»](#).

Задаёт высоту детекции в пикселях для Главного Трека. Когда размер детекции достигает указанного значения, FaceStream определяет лучшие снимки до окончания трека, а затем отправляет их в LP.

5 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](#)».

5.1 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.

5.1.1 logging

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

5.1.1.1 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.

5.1.1.2 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 14: 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
debug	Отладочная информация. Следует использовать данный тег только при отладке, а не при эксплуатации FS, т. к. выводится большое количество отладочной информации
estimator	Информация о статистике выполненных оценок.

Тег «estimator»

При включении тега «estimator» в логах FaceStream возвращается информация по размеру батчей, с которыми вызывались эstimаторы. Пример содержимого логов с включенным тегом «estimator»:

```
[I0609 15:48:03.779697    65 EstimatorStatistic.cpp:85] [estimator] Batch
    statistic for estimator HeadPoseEstimator
Total calls: 1311 total time: 191 ms.
sz  cnt
1   1311 (100.00%)
2    0 (0.00%)
3    0 (0.00%)
...
16   0 (0.00%)
```

В данном случае, статистика показывает следующее:

- всего было сделано 1311 вызова к эstimатору «HeadPoseEstimator», на которые ушло 191 миллисекунд;
- все вызовы (1311) были обработаны индивидуально, каждый в своем батче размером 1;
- батчи с размерами от 2 до 16 не были использованы.

Если эstimаторы не вызывались, то в логах FaceStream не будет выведено никакой информации.

5.1.1.3 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.

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

5.1.2 sending

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

5.1.2.1 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.

Подробное описание запросов см. в таблице.

Таблица 15: 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

5.1.2.2 request_timeout_ms

Данный параметр задает таймаут отправки и получения событий из/в LUNA PLATFORM в миллисекундах.

Значение по умолчанию — 30 секунд (30000 миллисекунд).

5.1.2.3 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.
- «crop» – использовать область, расширенную и обрезанную вокруг обнаруженного объекта.

Формат warp

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.

Формат crop

Crop-изображение это фрагмент изображения, полученный путём расширения и обрезки области вокруг обнаруженного объекта. Расширение области определяется новым параметром «[crop_factor](#)», увеличивающим размеры исходного прямоугольника, содержащего объект. Максимальный размер обрезанного изображения контролируется параметром новым параметром «[max_crop_size](#)», ограничивающим наибольшую сторону области. Если размер фрагмента превышает заданный параметр, то фрагмент будет масштабирован пропорционально по обеим координатам до нужного размера. Crop-изображение позволяет захватить больше контекста и детализации вокруг объекта по сравнению с биометрическим образцом.

Это позволяет более гибко управлять обработкой изображений. Отправка Crop-изображений имеет несколько ключевых преимуществ относительно других форматов:

- вместе с Crop-изображением передаются координаты лица как на Crop-изображении, так и на исходном изображении. Это позволяет избежать повторного запуска детектора лиц, обеспечивая наличие лица на изображении и экономя вычислительные ресурсы.
- Crop-изображение содержит больше контекста и деталей вокруг лица, что позволяет оценивать различные параметры, которые невозможно оценить на биометрическом образце. Например, эстимация Deepfake, Liveness и др., могут быть выполнены с большей точностью.
- сторонние сервисы и алгоритмы распознавания лиц часто работают лучше с Crop-изображениями, чем с биометрическими образцами. Это связано с тем, что Crop-изображения содержат больше информации и контекста, что улучшает точность распознавания и анализа.

5.1.2.4 crop_factor

Коэффициент увеличения размеров исходного прямоугольника, содержащего объект.

Этот параметр определяет, насколько шире и выше будет область обрезки по сравнению с исходным размером. Например, при «crop_factor» равном 2.2 размеры области обрезки будут увеличены в 2.2 раза по сравнению с исходными размерами. Это позволяет захватить больше контекста вокруг обнаруженного объекта.

5.1.2.5 max_crop_size

Максимальный размер обрезанного изображения.

Этот параметр ограничивает наибольшую сторону области обрезки, чтобы она не превышала заданное значение. Если обрезанное изображение превышает «max_crop_size», оно будет масштабировано до допустимого размера, сохраняя пропорции. Это гарантирует, что обрезанные изображения не будут слишком большими и остаются в пределах заданных размеров.

5.1.2.6 send_source_frame

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

Исходный кадр можно сохранять как в бакет сервиса LUNA Image Store, так и в ресурс «/images» сервиса LUNA API. Способ сохранения исходного кадра задается с помощью указания соответствующего адреса в параметре «frame_store».

Для коллекции лучших снимков («face_bestshots_to_send» или «body_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.

Если предполагается сохранять исходный кадр в бакет сервиса LUNA Image Store, то кадр будет отправлен в LP за некоторое время до отправки лучшего снимка. Сначала 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.

Пример из тела ответа события об успешном сохранении в бакет:

```
"events": [  
  "detections": [  
    "image_origin": "/1/buckets/source-images/images/83c383d8-8af4-406c-8ff8-76eb389e61c8"  ]  
]
```

Если предполагается сохранять исходный кадр в ресурс «/images», то кадр будет отправлен вместе с лучшим снимком с помощью схемы «multipart/form-data» запроса на генерацию события LUNA PLATFORM.

Пример из тела ответа события об успешном сохранении в ресурс «/images» сервиса LUNA API:

```
"events": [  
  "detections": [  
    "image_origin": "/6/images/19c5c40a-04e4-4f2b-811c-0f3bb0003359"  ]  
]
```

Если же в поле «image_origin» содержится адрес вида «/6/samples/faces/c93f39f1-809c-4583-b386-c166сесас333», то это значит, что речь идет об адресе сохранения лучшего снимка, а не исходного изображения.

5.1.2.7 size_source_frame

Параметр изменяет ширину исходного кадра до заданного значения. Допустимый диапазон — [0...1024]. Значение «0» означает, что ширина не будет меняться. Дробная часть задаваемого числа игнорируется.

FaceStream отправляет в LUNA PLATFORM заголовок X-Luna-Meta-rescale, который содержит коэффициент масштабирования исходного изображения в форме числа с плавающей запятой.

Коэффициент масштабирования позволяет вычислить реальные координаты отмасштабированного исходного изображения. Для этого необходимо разделить размер полученного изображения на полученный коэффициент масштабирования.

Примечание. Реальный размер рассчитывается с абсолютной погрешностью менее 1 / (2 * коэффициент масштабирования).

Например, полученное изображение имеет размер 200x113, а коэффициент масштабирования равен 0.104399, тогда реальный размер исходного изображения равен 1916x1082 (200/0.104399=1916, 113/0.104399=1082).

Коэффициент масштабирования можно найти в LUNA PLATFORM:

- в заголовке ответа на запрос «get image» к сервису LUNA API. Запрос должен быть выполнен с параметром запроса with_meta = 1
- в файле «image_id.meta.json», расположенном рядом с исходным изображением в бакете исходных изображений хранилища LUNA Image Store

Заголовок X-Luna-Meta-rescale отправляется только если включена отправка исходного кадра (параметр send_source_frame = 1). В обработчике LUNA PLATFORM также должна быть включена политика сохранения исходного изображения.

Если для исходного кадра не выставлено масштабирование (параметр size_source_frame = 0), то значение заголовка ответа X-Luna-Meta-rescale будет равно 1.

5.1.2.8 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.

5.1.2.9 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 спецификации OpenAPI LUNA PLATFORM). По набором детекций тела человека можно определить его путь.

Детекция, у которой есть лучший снимок обязательна к отправке.

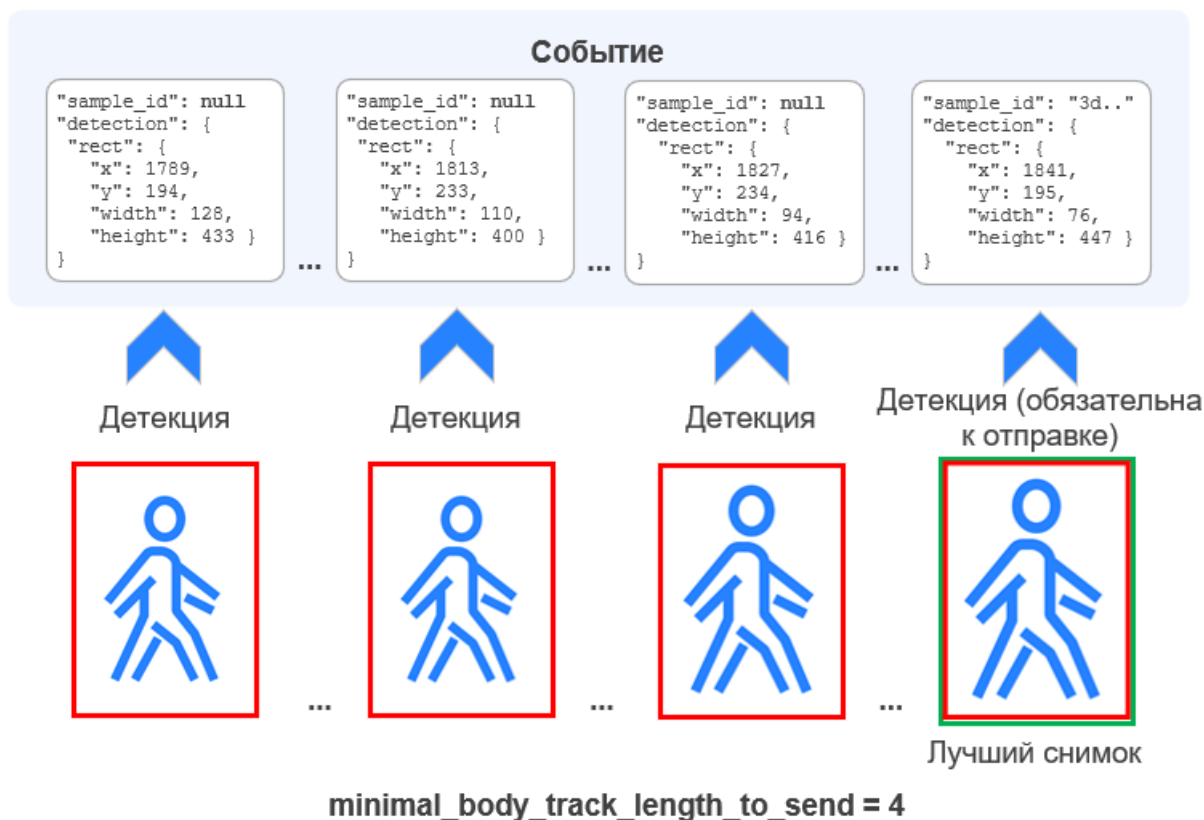


Рис. 18: Отправка детекций

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 количество детекций будет равно количеству лучших снимков.

5.1.2.10 `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.

5.1.2.11 `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 `«face_bestshots_to_send»` или `«body_bestshots_to_send»`.

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

5.1.2.12 `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.

5.1.3 `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»](#).

5.1.3.1 origin

Адрес и порт сервера, где запущен сервис LUNA Streams.

5.1.3.2 api_version

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

Актуальную версию API всегда можно найти в документации сервиса API.

5.1.4 lunaplatform

Данный раздел описывает настройки подключения к LUNA PLATFORM (сервису API) для выполнения анализа видео.

5.1.4.1 origin

Адрес и порт сервера, где запущен сервис LUNA API.

5.1.4.2 api_version

В параметре задаётся версия API сервиса LUNA API. На настоящий момент поддерживается 6 версия API.

5.1.4.3 max_number_streams

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

5.1.4.4 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.

5.1.4.5 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.

5.1.4.6 [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_OBSOLETEING_PERIOD](#)», задаваемого в настройках сервиса LUNA Streams.

Значение по умолчанию — 10 секунд.

5.1.5 [performance](#)

5.1.5.1 [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.

5.1.5.2 [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»).

5.1.5.3 [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.

5.1.6 fps_floor

Параметр предназначен для управления количеством анализируемых кадров в видеопотоке, что позволяет уменьшить нагрузку на систему за счет пропуска некоторых кадров. Это может быть полезно, когда необходимо снизить требования к вычислительным ресурсам, сохраняя при этом приемлемую точность анализа.

Количество пропущенных кадров рассчитывается как разница между частотой кадров видео потока и значением «fps_floor». Пропущенные кадры распределяются равномерно в течение каждой секунды.

Если значение «fps_floor» равно нулю или отрицательное, FaceStream проигнорирует настройку, и количество обрабатываемых кадров останется неизменным.

Например, при значении «fps_floor» = 15 для видеопотока с частотой 25 fps FaceStream будет пропускать 10 кадров в секунду (25 - 15). Эти 10 кадров будут равномерно распределены по времени, чтобы сохранить целостность воспроизведения видео и избежать резких скачков.

5.1.7 monitoring

В данной группе параметров задаются настройки мониторинга.

5.1.7.1 storage_type

Данный параметр позволяет выбрать тип хранения данных для мониторинга. На данный момент можно использовать только базу данных Influx.

5.1.7.2 send_data

Данный параметр включает отправку мониторинга. По умолчанию отключен.

5.1.7.3 organization

Данный параметр задает рабочую область InfluxDB 2.x.

Рабочая область задается с помощью передачи соответствующего аргумента при старте контейнера InfluxDB.

5.1.7.4 bucket

Данный параметр задает бакет InfluxDB 2.x.

Бакет задается с помощью передачи соответствующего аргумента при старте контейнера InfluxDB. Его можно также создать с помощью пользовательского интерфейса Influx.

5.1.7.5 token

Параметр задаёт токен аутентификации InfluxDB 2.x.

Токен задается с помощью передачи соответствующего аргумента при старте контейнера InfluxDB.

5.1.7.6 origin

Параметр задаёт IP-адрес и порт 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.

```
"origin": "http://127.0.0.1:8086/"
```

5.1.7.7 flushing_period

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

По умолчанию используется 1 секунда.

5.2 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»](#)).

5.2.1 use-face-detector и use-body-detector

Данные параметры позволяют включать и отключать выполнение детекций лиц и тел. Можно включить одновременную детекцию.

См. раздел [«Совместный режим детектирования лиц и тел»](#) для более подробной информации.

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" />
```

5.2.2 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" />
```

5.2.3 detector-scaling

Параметр «detector-scaling» позволяет масштабировать кадр перед обработкой. Размер кадра для масштабирования задается в параметре «scale-result-size».

См. подробную информацию о масштабировании в разделе «Масштабирование кадра».

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

5.2.4 scale-result-size

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

Для работы данного параметра необходимо включить параметр «detector-scaling».

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.

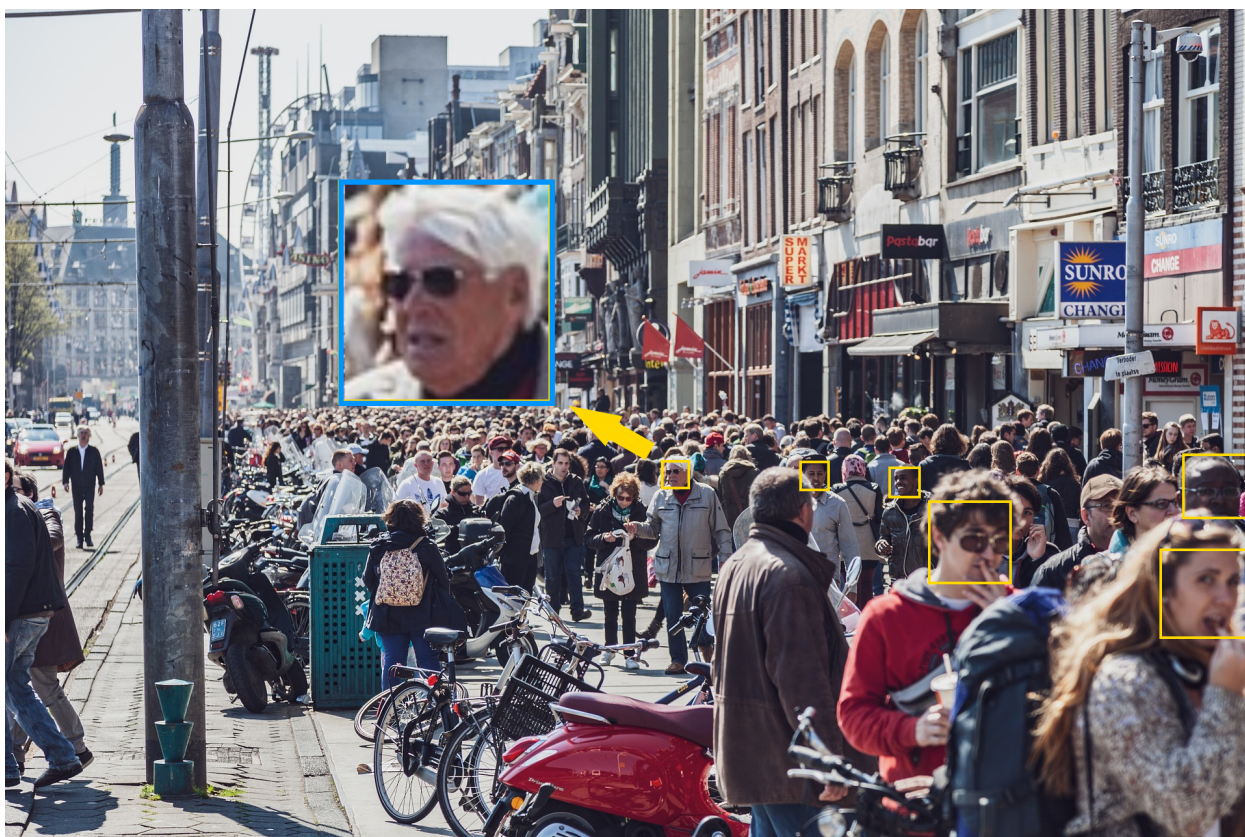


Рис. 19: Детекции на кадре с разрешением 1920X1080

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

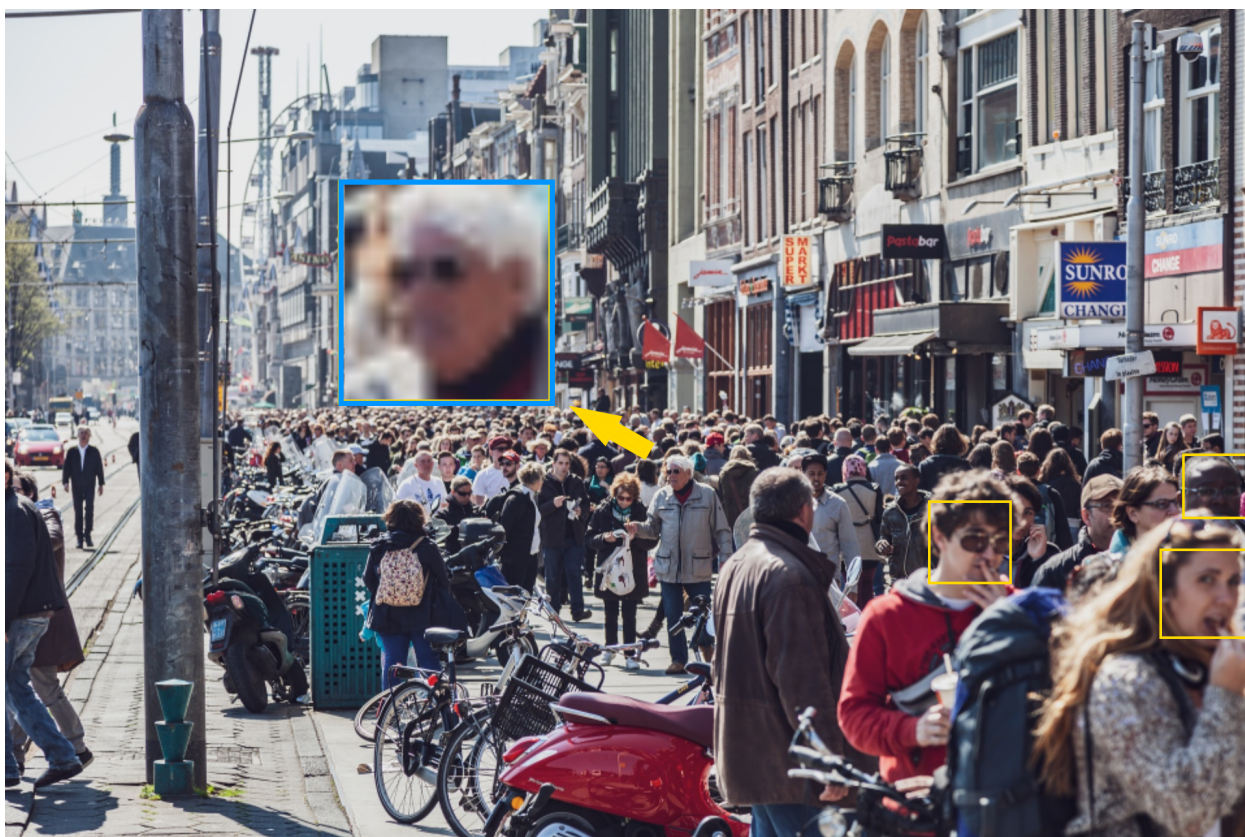


Рис. 20: Детекции на кадре с разрешением 960X640

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" />
```

5.2.5 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" />
```

5.2.6 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" />
```

5.2.7 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" />
```

5.2.8 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" />
```

5.2.9 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" />
```

5.2.10 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" />
```

5.2.11 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 выставлять 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" />
```

5.3 Настройки LUNA Streams

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

5.3.1 LUNA_STREAMS_DB

5.3.1.1 db_type

Параметр задает тип базы данных. Доступны следующие типы:

- «postgres» — тип базы данных PostgreSQL
- «oracle» — тип базы данных Oracle

Формат задания настройки — string.

Значение по умолчанию — postgres.

5.3.1.2 db_host

Параметр задает имя сервера (хост) базы данных.

Формат задания настройки — string.

Значение по умолчанию — 127.0.0.1.

5.3.1.3 db_port

Параметр задает порт базы данных.

Порт по умолчанию для типа «postgres» — 5432.

Порт по умолчанию для типа «oracle» — 1521.

Формат задания настройки — string.

Значение по умолчанию — 5432.

5.3.1.4 db_user

Параметр задает имя пользователя базы данных.

Формат задания настройки — string.

Значение по умолчанию — luna.

5.3.1.5 db_password

Параметр задает имя пользователя базы данных.

Формат задания настройки — string.

Значение по умолчанию — luna.

5.3.1.6 db_name

Параметр задает имя базы данных для типа «postgres» или имя SID для типа «oracle».

Формат задания настройки — string.

Значение по умолчанию — luna_streams.

5.3.1.7 db_settings > connection_pool_size

Параметр задает размер пула соединений к БД.

Формат задания настройки — string.

Значение по умолчанию — 10.

5.3.1.8 dsn

Параметр задает строку подключения DSN для соединения с БД.

Формат задания настройки — string.

DSN — это строка подключения, которая идентифицирует и указывает на источник данных (базу данных), к которому нужно установить соединение.

В строке DSN могут задаваться такие настройки, как множественные хосты, аутентификационные данные, порт и другие параметры.

Настройки зависят от типа БД. Множественные хосты поддерживаются только с PostgreSQL.

По умолчанию параметр «dsn» не отображается во вкладке «Settings» в Configurator. Проверить список всех доступных параметров для группы настроек можно на вкладке «Limitations».

Пример:

```
{
  "dsn": "luna:luna@postgres01:5432,postgres02:5432/luna_streams?some_option=
    some_value"
  "db_settings": {
    "connection_pool_size": 5
  }
}
```

Здесь:

- «luna:luna» — имя пользователя и пароль для подключения к PostgreSQL;
- «@postgres01:5432,postgres02:5432» — список хостов и портов, разделенных запятыми. Это означает, что сервис будет пытаться подключиться к первому хосту («postgres01») на порту

5432. Если это не удастся, она попытается подключиться ко второму хосту («postgres02») также на порту 5432;

- «/luna_streams» — БД сервиса Streams;
- «?some_option=some_value» — опциональные параметры для подключения.

При необходимости можно комбинировать строку DSN и классические настройки, однако строка DSN является более приоритетной. Можно частично заполнить строку DSN (например, «postgres01,postgres02/luna_streams»), и тогда недостающие параметры будут заполнены из значений параметров «db_host», «db_port», «db_name», «db_user» и «db_password».

При запуске сервис создаст пул подключений к одному из доступных хостов DSN. В случае возникновения проблем с установлением соединения после нескольких неудачных попыток, сервис снова попытается настроить пул подключений к любому из доступных хостов DSN.

5.3.2 LUNA_LICENSES_ADDRESS

Данная группа параметров задаёт настройки подключения к сервису LUNA Licenses.

5.3.2.1 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.

Формат задания настройки — string.

Значение по умолчанию — http://127.0.0.1:5120.

5.3.2.2 api_version

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

Формат задания настройки — integer.

Значение по умолчанию — 1.

5.3.3 LUNA_STREAMS_LOGGER

Данная группа параметров задает настройки логирования сервиса Streams.

5.3.3.1 log_level

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

Формат задания настройки — `string`.

Значение по умолчанию — `INFO`.

5.3.3.2 `log_time`

Параметр задает формат времени, используемый в записях лога. Доступны следующие значения:

- «LOCAL» — отображает местное время системы, на которой выполняется запись логов;
- «UTC» — отображает координированное всемирное время, которое является стандартом времени и не зависит от местной временной зоны или сезонных изменений времени.

Формат задания настройки — `string`.

Значение по умолчанию — `LOCAL`.

5.3.3.3 `log_to_stdout`

Параметр позволяет отправлять логи в стандартный вывод (`stdout`).

Формат задания настройки — `boolean`.

Значение по умолчанию — `true`

5.3.3.4 `log_to_file`

Параметр позволяет сохранять логи в файл. Директория с файлами логов указывается в параметре «`folder_with_logs`».

Формат задания настройки — `boolean`.

Значение по умолчанию — `false`.

5.3.3.5 `folder_with_logs`

Параметр задает директорию, в которой хранятся логи. Относительный путь начинается с каталога с приложением.

Для использования данного параметра требуется включить параметр «`log_to_file`».

Формат задания настройки — `string`.

Значение по умолчанию — `./`

Пример:

```
"folder_with_logs": "/srv/logs"
```


5.3.3.6 max_log_file_size

Параметр задает максимальный размер файла лога в МБ перед выполнением его ротации (0 — не использовать ротацию).

Для использования данного параметра требуется включить параметр [«log_to_file»](#).

Формат задания настройки — integer.

Значение по умолчанию — 1024

5.3.3.7 multiline_stack_trace

Параметр включает многострочную трассировку стека в логах. Когда параметр включен, информация о стеке вызовов записывается в логах так, что каждый фрейм стека помещается на отдельной строке, что улучшает читаемость. Если параметр выключен, информация о стеке вызовов записывается на одной строке, что может сделать логи менее удобными для анализа.

Формат задания настройки — boolean.

Значение по умолчанию — true.

5.3.3.8 format

Параметр определяет формат выводимых логов. Доступны следующие значения:

- «default» — стандартный формат вывода логов LUNA PLATFORM
- «json» — вывод логов в формате json
- «ecs» — вывод логов в формате ECS (Elastic Common Schema)

При использовании значения «ecs» будут использоваться следующие поля:

- «http.response.status_code» — содержит код состояния ответа HTTP (200, 404, 500 и т.д.);
- «http.response.execution_time» — содержит информацию о времени, затраченном на выполнение запроса и получение ответа;
- «http.request.method» — содержит метод HTTP-запроса (GET, POST, PUT и т.д.);
- «url.path» — содержит путь в URL-адресе запроса;
- «error.code» — содержит код ошибки, если запрос завершается с ошибкой.

Формат задания настройки — string.

Значение по умолчанию — default.

5.3.4 LUNA_MONITORING

Данная группа параметров задает настройки мониторинга.

См. подробную информацию о мониторинге в разделе [«Мониторинг»](#).

5.3.4.1 `storage_type`

Тип хранилища для хранения данных мониторинга.

В настоящее время доступно только БД Influx.

Формат задания настройки — `string`.

Значение по умолчанию — `influx`.

5.3.4.2 `send_data_for_monitoring`

Параметр позволяет включить или отключить отправку данных для мониторинга в InfluxDB.

Формат задания настройки — `integer`.

Значение по умолчанию — `1`.

5.3.4.3 `use_ssl`

Параметр позволяет использовать HTTPS для подключения к InfluxDB.

Формат задания настройки — `integer`.

Значение по умолчанию — `0`.

5.3.4.4 `organization`

Параметр задает рабочую область InfluxDB 2.x.

Формат задания настройки — `string`.

Значение по умолчанию — `luna`.

5.3.4.5 `token`

Параметр задает аутентификации InfluxDB 2.x.

Формат задания настройки — `string`.

5.3.4.6 `bucket`

Параметр задаёт имя бакета InfluxDB 2.x.

Формат задания настройки — `string`.

Значение по умолчанию — `luna_monitoring`.

5.3.4.7 host

Параметр задаёт IP-адрес InfluxDB.

Формат задания настройки — `string`.

Значение по умолчанию — `127.0.0.1`.

5.3.4.8 port

Параметр задаёт порт InfluxDB.

Формат задания настройки — `string`.

Значение по умолчанию — `8086`.

5.3.4.9 flushing_period

Параметр задает частоту отправки данных мониторинга в InfluxDB.

Формат задания настройки — `integer` (секунды).

Значение по умолчанию — `1`.

5.3.5 Группа параметров LUNA_SERVICE_METRICS

Данная группа параметров включает и настраивает сбор метрик в формате Prometheus.

См. подробную информацию о мониторинге в разделе [«Мониторинг»](#).

5.3.5.1 enabled

Параметр включает сбор метрик.

Если сбор метрик отключен, то запрос к ресурсу `/metrics` вернет соответствующее сообщение.

Формат задания настройки — `boolean`.

Значение по умолчанию — `false`.

5.3.5.2 metrics_format

Параметр задает формат метрик.

На данный момент поддерживается только формат Prometheus.

См. [официальную документацию Prometheus](#) для более подробной информации.

Формат задания настройки — `string`.

Значение по умолчанию — `prometheus`.

5.3.5.3 extra_labels

Параметр задает пользовательские типы лейблов.

Формат задания настройки — `label_name=label_value`.

Значение по умолчанию не задано.

5.3.6 LUNA_STREAMS_HTTP_SETTINGS

В данной группе параметров содержатся настройки, отвечающие за обработку HTTP-подключений. См. подробную информацию по следующей ссылке: <https://sanic.dev/en/guide/deployment/configuration.html#builtin-values>

5.3.6.1 request_timeout

Параметр задает продолжительность времени между моментом, когда новое открытое TCP-соединение передается на сервер, и моментом, когда получен весь HTTP-запрос.

Формат задания настройки — `integer` (секунды).

Значение по умолчанию — 60.

5.3.6.2 response_timeout

Параметр задает продолжительность времени между моментом, когда сервер передает HTTP-запрос приложению, и моментом, когда HTTP-ответ отправляется клиенту.

Формат задания настройки — `integer` (секунды).

Значение по умолчанию — 600.

5.3.6.3 request_max_size

Параметр задает максимальный размер запроса.

Формат задания настройки — `integer` (байты).

Значение по умолчанию — 1073741824.

5.3.6.4 keep_alive_timeout

Параметр задает тайм-аут поддержания активности HTTP.

Формат задания настройки — `integer` (секунды).

Значение по умолчанию — 15.

5.3.7 LUNA_STREAMS_LOGS_CLEAR_INTERVAL

В данной группе параметров содержатся настройки, отвечающие автоматическое удаление логов потоков. .

Автоматическое удаление логов помогает очистить таблицу «log» базы данных LUNA Streams от большого количества ненужных логов.

См. подробное описание с примерами в разделе «[Автоматическое удаление логов потоков](#)».

5.3.7.1 active

Параметр включает или отключает автоматическое удаление логов.

Формат задания настройки — `boolean`.

Значение по умолчанию — `0`.

5.3.7.2 interval

Параметр задает интервал удаления логов. Логи, старше чем данный интервал, будут удалены.

Интервал должен задаваться в паре с [типом интервала](#). Например, «3» (интервал) + день (тип интервала).

Формат задания настройки — `integer`.

Значение по умолчанию — `7`.

5.3.7.3 interval_type

Параметр задает один из следующих типов интервала:

- «weeks» (недели);
- «days» (дни);
- «hours» (часы);
- «minutes» (минуты);
- «seconds» (секунды).

Тип интервала должен задаваться в паре с интервалом. Например, «3» ([интервал](#)) + день (тип интервала).

Формат задания настройки — `string`.

Значение по умолчанию — `days`.

5.3.7.4 check_interval

Параметр задает частоту проверки логов на удаление в секундах.

Формат задания настройки — `integer`.

Значение по умолчанию — 180.

5.3.8 Прочие

5.3.8.1 `stream_worker_async_lock_timeout`

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

Формат задания настройки — `number` (секунды в диапазоне (0, 1]).

Значение по умолчанию — 0.1.

5.3.8.2 `stream_status_obsoleting_period`

Параметр задаёт период устаревания состояния потока. За данный период времени «рабочий процесс» FaceStream должен передать отчёт в LUNA Streams. В противном случае статус потока будет изменен на «restart», а запоздалый отчёт будет отвергнут.

Формат задания настройки — `number` (секунды в диапазоне (0, 86400]).

Значение по умолчанию — 20.

5.3.8.3 `luna_streams_active_plugins`

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

Формат задания настройки — `array > string`.

Значение по умолчанию — [].

5.3.8.4 `storage_time`

Параметр задаёт формат времени, используемый для записей в базе данных. Доступны следующие значения:

- «LOCAL» — отображает местное время системы, на которой выполняется запись логов;
- «UTC» — отображает координированное всемирное время, которое является стандартом времени и не зависит от местной временной зоны или сезонных изменений времени.

Формат задания настройки — `string`.

Значение по умолчанию — LOCAL.

6 Настройки FaceEngine

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

Конфигурирование параметров FaceEngine осуществляется только с помощью редактирования конфигурационного файла «faceengine.conf». Конфигурационный файл монтируется в контейнер FaceStream при его запуске (см. команду запуска FaceStream в руководстве по установке).

В данном разделе задаются параметры, касающиеся глубокой настройки распознавания лица и тела. Изменять параметры рекомендуется только по согласованию с сотрудниками VisionLabs.

6.1 FaceDetV3

В данном параметре задаются настройки для детектора лиц.

Примечание. Обратите внимание, что данная группа параметров используется только при включенном параметре «[use-face-detector](#)». При работе с телами будут использоваться параметры из группы параметров «[HumanDetector](#)».

6.1.1 minFaceSize

Параметр задает коэффициент масштабирования исходного изображения перед началом обработки.

Размер изображений изменяется на $\text{minFaceSize}/20$ раз. Например, если значение равно $\text{minFaceSize}=50$, то размер изображения будет изменен в 2.5 раза ($\text{minFaceSize}=50/20$).

См. подробную информацию о масштабировании в разделе «[Масштабирование кадра](#)».

6.2 HumanDetector

В данном параметре задаются настройки для детектора тел.

Примечание. Обратите внимание, что данная группа параметров используется только при включенном параметре «[use-body-detector](#)». При работе с лицами будут использоваться параметры из группы параметров «[FaceDetV3](#)».

6.2.1 imageSize

Параметр задает максимальный размер кадра после масштабирования по наибольшей из сторон кадра.

Данный параметр работает аналогично параметру «[scale-result-size](#)» из настроек TrackEngine.

См. подробную информацию о масштабировании в разделе «[Масштабирование кадра](#)».

7 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.

7.1 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.

7.2 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 16: 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

7.3 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. 21: 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. 22: Задание тега

Нельзя создавать теги для записей по умолчанию.

8 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.5.2.3 \
--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 \
```

8.1 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».

9 Пользовательский интерфейс сервиса LUNA Streams

Для сервиса LUNA Streams доступен пользовательский интерфейс.

Интерфейс можно открыть в браузере, указав адрес и порт сервиса LUNA Streams:

`<streams_server_address>:<streams_server_port>`

Порт сервиса LUNA Streams по умолчанию — 5160.

Рекомендуется ознакомиться с разделом [«Взаимодействие FaceStream с LUNA Streams»](#) перед ознакомлением с данной главой.

Пользовательский интерфейс сервиса содержит три вкладки — «Streams», «Groups» и «Queue».

- «Streams» — вкладка, в которой отображаются [статусы потоков](#) и их предварительный просмотр, есть возможность настраивать параметры видеопотока для каждого из источников.
- «Groups» — вкладка, в которой отображаются [группы потоков](#).
- «Queue» — вкладка в которой отображаются потоки, находящиеся в [очереди на обработку](#).

Общий вид доступных вкладок представлен на рисунке ниже:

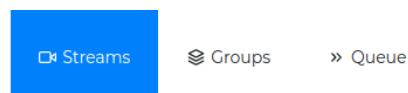


Рис. 23: Доступные вкладки

9.1 Вкладка «Streams»

Вкладка «Streams» предназначена для отображения всех потоков, их предварительного просмотра, идентификатора, названия, статуса, описания, группы и настройки параметров видеопотока для каждого источника.

Общий вид вкладки «Streams» представлен на рисунке ниже:

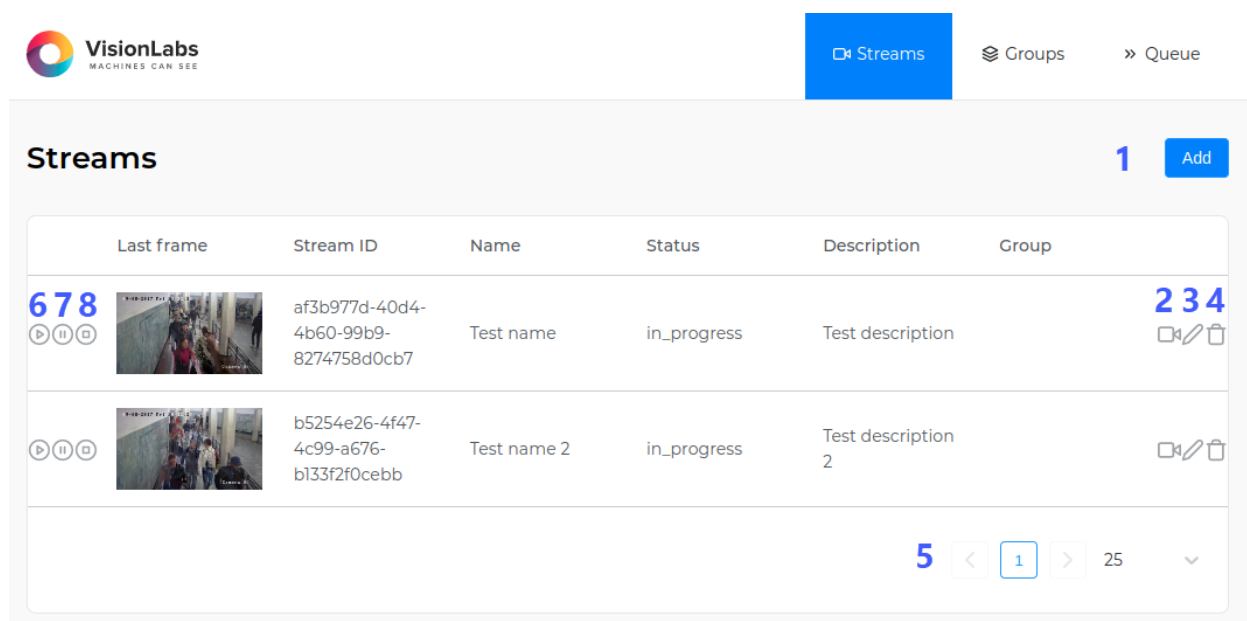


Рис. 24: Общий вид вкладки «Streams»


Вкладка «Streams» содержит следующие элементы:

- кнопка «Add» — кнопка для добавления потока (1);
- перечень потоков:
 - «Last frame» — последний кадр видеопотока (предварительный просмотр);
 - «Stream ID» — идентификатор видеопотока;
 - «Name» — название видеопотока;
 - «Status» — текущий статус видеопотока;
 - «Description» — дополнительная информация о видеопотоке;
 - «Group» — название группы, к которой привязан видеопоток. Привязать видеопоток к группе или отвязать от нее можно в разделе «Groups»;
 - кнопка для просмотра видеопотока (2);
 - кнопка для редактирования параметров видеопотока (3);
 - кнопка для удаления видеопотока (4);
 - количество отображаемых на странице потоков задается переключателем в нижнем правом углу страницы. Всего может быть 10, 25, 50 или 100 видеопотоков на одной странице (5);
 - кнопки для управления обработкой потока:
 - * кнопка «Play» для запуска обработки потока (6) (отправляет запрос на обработку потока, поток распределяется на определенный экземпляр FaceStream и тот начинает его обрабатывать);
 - * кнопка «Pause» для паузы в обработке потока, например, для экономии ресурсов (7) (приостанавливает процесс обработки потока, но поток остается закрепленным за

- тем же экземпляром FaceStream);
- * кнопка «Stop» для остановки обработки потока (8) (останавливает процесс обработки потока, поток больше не закреплен за тем же экземпляром FaceStream).

9.1.1 Создание потока

Для добавления потока нажмите на кнопку «Add», после чего откроется форма «Create stream» для указания параметров.



Streams

Groups

» Queue

Create stream

Create

General stream parameters

Account ID

Stream name

Description

Group

Status

Stream data

Type

Full path to the source

ROI coordinates

DROI coordinates

Rotation angle of the image from the source

Frame width

Endless

Sending parameters

Frame analysis period after which the best shot will be sent

Wait duration between track analysis periods

Track analysis and waiting period duration measure

Number of frames that the user sets to receive from the track or certain periods of this track

Send only full set

Delete bestshot and detection data

Use Primary Track

Healthcheck parameters

Maximum number of stream errors to reconnect to the stream

Error count period duration (in seconds)

Time between reconnection attempts (in seconds)

Рис. 25: Создание потока

Необходимо указать значения параметров потока и нажать кнопку «Create» в правом верхнем углу экрана.

В таблицах ниже приведено соответствие между описанием в пользовательском интерфейсе и названием соответствующих параметров.

9.1.1.1 Группа «General stream parameters»

В данной группе отражены общие параметры создания потока.

Таблица 17: Группа «General stream parameters»

Описание параметра	Название в настройках
Account ID	Параметр «account_id»
Stream name	Параметр «name»
Description	Параметр «description»
Group	Параметр «group»
Status	Статус потока

9.1.1.2 Группа «Stream data»

В данной группе отражены основные параметры для работы с видеопотоком/видеофайлом/изображениями.

Таблица 18: Группа «Stream data»

Описание параметра	Название в настройках
Type	Параметр «type»
Full path to the source	Параметр «reference»
ROI coordinates	Параметр «roi»
DROI coordinates	Параметр «droi»
Rotation angle of the image from the source	Параметр «rotation»
Frame width	Параметр «preferred_program_stream_frame_width»
Endless	Параметр «endless»
Mask	Параметр «mask»

9.1.1.3 Группа «Stream handler parameters»

Данная группа параметров определяет параметры обработчика, создаваемого в LUNA PLATFORM, с помощью которого будут обрабатываться потоки. Для лиц и тел следует использовать различ-

ные обработчики. Обработчик должен быть создан в LP 5 заранее.

Таблица 19: Группа «Stream handler parameters»

Описание параметра	Название в настройках
Handler URL	Параметр «origin»
API version	Параметр «api_version»
ID of the handler for the best frames	Параметр «handler_id» группы «bestshot_handler»
Authorization (Token)	Токен из группы параметров «authorization»

9.1.1.4 Группа «Geoposition»

Данная группа параметров включает информацию о расположении источника видеопотока.

Таблица 20: Группа «Geoposition»

Параметры	Название в настройках
City, Area, District, Street, House number, Longitude (in degrees), Latitude (in degrees)	Группа параметров «location»

9.1.1.5 Группа «Autorestart»

Данная группа параметров позволяет настроить автоматический перезапуск потока.

Таблица 21: Группа «Geoposition»

Описание параметра	Название в настройках
Autorestart	Параметр «restart»
Attempt count	Параметр «attempt_count»
Autorestart delay (in seconds)	Параметр «delay»

9.1.1.6 Группа «Sending parameters»

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

Таблица 22: Группа «Sending parameters»

Описание параметра	Название в настройках
Frame analysis period after which the best shot will be sent	Параметр « time_period_of_searching »
Wait duration between track analysis periods	Параметр « silent_period »
Track analysis and waiting period duration measure	Параметр « type »
Number of frames that the user sets to receive from the track or certain periods of this track	Параметр « number_of_bestshots_to_send »
Send only full set	Параметр « send_only_full_set »
Delete best shot and detection data	Параметр « delete_track_after_sending »

9.1.1.7 Группа «Use Primary Track»

Данная группа параметров предназначена для работы с системами контроля управления доступом (СКУД, турникеты на входах) для упрощения контроля и внедрения технологии распознавания лиц при входе на охраняемую территорию. Данная группа параметров используется только для работы с лицами.

Таблица 23: Группа «Use Primary Track»

Описание параметра	Название в настройках
Use Primary Track	Параметр « use_primary_track_policy »
Minimum detection size for Primary Track mode	Параметр « best_shot_min_size »
Size of detection for the main track	Параметр « best_shot_proper_size »

9.1.1.8 Группа «Healthcheck parameters»

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

Таблица 24: Группа «Healthcheck parameters»

Описание параметра	Название в настройках
Maximum number of stream errors to reconnect to the stream	Параметр « max_error_count »
Error count period duration (in seconds)	Параметр « period »
Time between reconnection attempts (in seconds)	Параметр « retry_delay »

9.1.1.9 Группа «Liveness parameters»

Данная группа параметров используется проверки на Liveness, т.е. для проверки наличия живого человека в кадре и предотвращает использование распечатанного фото или фото с телефона для прохождения проверки.

Таблица 25: Группа «Liveness parameters»

Описание параметра	Название в настройках
Check RGB ACS Liveness	Параметр « use_mask_liveness_filtration »
Check FlyingFaces Liveness	Параметр « use_flying_faces_liveness_filtration »
Which track frames will be checked for Liveness	Параметр « liveness_mode »
Number of frames in the track for Liveness check when liveness-mode is enabled	Параметр « number_of_liveness_checks »
Threshold value at which the system will consider that there is a real person in the frame	Параметр « liveness_threshold »
Livenesses weights (RGB ACS, FlyingFaces)	Параметр « livenesses_weights »
Number of background frames that are used for the corresponding checks	Параметр « mask_backgrounds_count »

9.1.1.10 Группа «Filtering parameters»

Данная группа параметров описывает объекты фильтрации изображений и отправки результирующих лучших кадров.

Таблица 26: Группа «Liveness parameters»

Описание параметра	Название в настройках
Threshold value to filter detections	Параметр «min_score»
Head rotation angle threshold (to the left or right, yaw)	Параметр « detection_yaw_threshold »
Head tilt angle threshold (up or down, pitch)	Параметр « detection_pitch_threshold »
Head tilt angle threshold (to the left or right, roll)	Параметр « detection_roll_threshold »
Number of frames used to filter photo images by the angle of tilt of the head	Параметр « yaw_number »
Number of frames the system must collect to analyze head yaw angle	Параметр « yaw_collection_mode »
Mouth overlap threshold (minimum mouth visibility)	Параметр « mouth_occlusion_threshold »
Minimum body detection size	Параметр « min_body_size_threshold »

9.1.1.11 Группа «Additional parameters»

Данная группа параметров предназначена для детальной настройки обработки видеопотока.

Таблица 27: Группа «Liveness parameters»

Описание параметра	Название в настройках
Frame processing	Параметр « frame_processing_mode »
Number of threads for video decoding	Параметр « ffmpeg_threads_number »
Maximum FPS for video processing	Параметр « real_time_mode_fps »

После сохранения настроек для вновь созданного потока появится сообщение Source <stream_id> was successfully created.

9.1.2 Редактирование потока

Для редактирования параметров потока необходимо нажать кнопку «Edit».

Форма редактирования потока аналогична форме создания потока.

Измените необходимые параметры и нажмите кнопку «Save» в правом верхнем углу формы.

После сохранения настроек параметров потока появится сообщение `Source <stream_id> was successfully updated.`

9.1.3 Удаление потока

Для удаления потока необходимо нажать кнопку «Delete» или в форме редактирования потока нажать кнопку «Delete» во вкладке «Streams».

Во всплывающем окне необходимо подтвердить действие — нажать кнопку «Delete» или отменить действие через кнопку «Cancel».

После нажатия кнопки «Delete» появится сообщение об удалении потока.

9.2 Вкладка «Groups»

Потоки можно группировать. Группировка предназначена для объединения потоков с несколькими камерами в логические группы (например, по территориальному признаку).

См. подробную информацию в разделе [«Группировка потоков»](#).

При создании потока, его можно добавить только в одну группу.

Вкладка «Groups» содержит таблицу, которая отображает название групп, идентификаторы групп, описания, даты создания, ID аккаунта в LUNA PLATFORM и настройки параметров групп.

Вкладка «Groups» представлена на рисунке ниже:

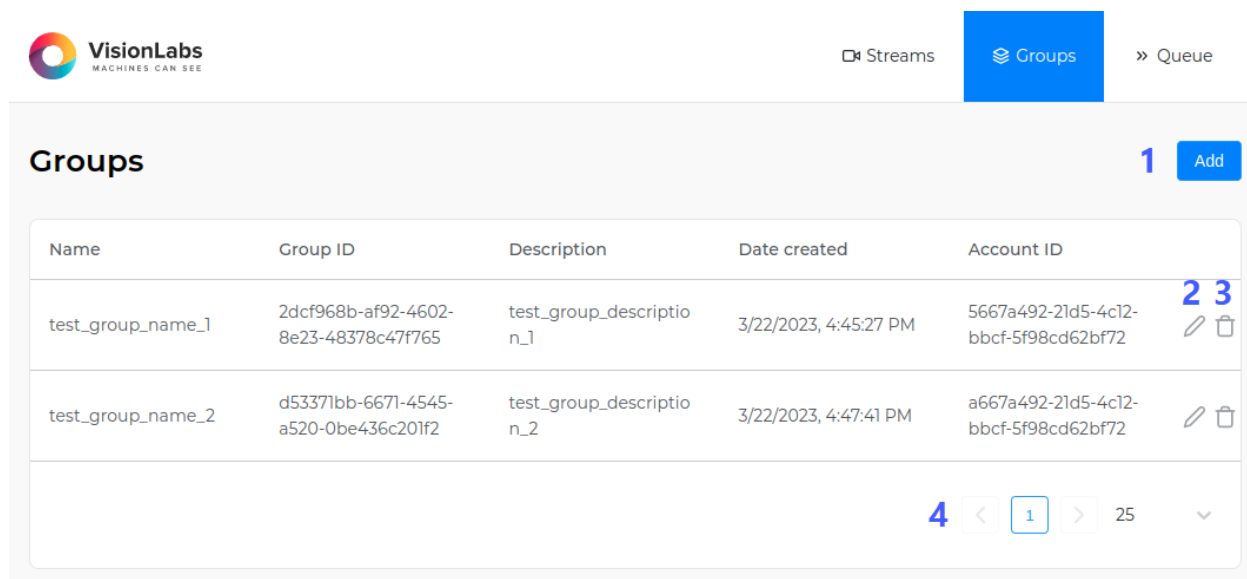


Рис. 26: Вкладка «Groups»

Вкладка «Groups» содержит следующие элементы и параметры группы:

- кнопка «Add» — кнопка для создания группы (1);
- перечень групп и их параметров:
 - «Name» — название группы;
 - «Group ID» — идентификатор группы, присваиваемый LUNA Streams;
 - «Description» — дополнительная пользовательская информация о группе;
 - «Date created» — дата и время создания группы;
 - «Account ID» — идентификатор аккаунта в LUNA PLATFORM 5;
- кнопка редактирования группы (2);
- кнопка удаления группы (3);
- количество отображаемых на странице групп задается переключателем в нижнем правом углу страницы. Всего может быть 10, 25, 50 или 100 групп на одной странице (4).

Процесс создания, редактирования и удаления группы аналогичен вышеописанным процессам для потока.

9.2.1 Привязка потока к группе

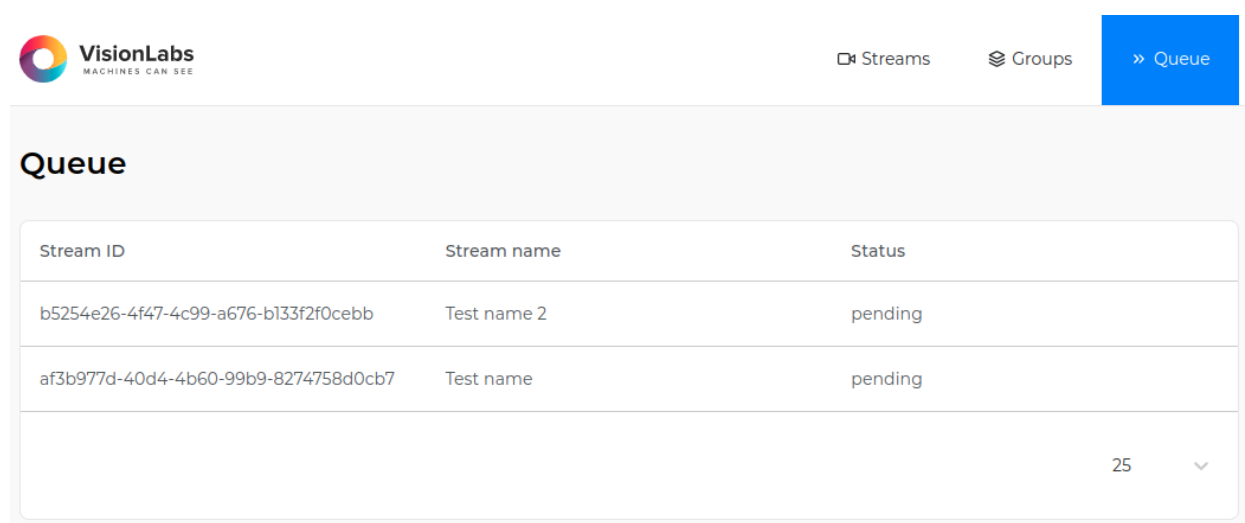
Чтобы привязать поток к группе, в разделе «Streams» в строке потока нажмите кнопку для редактирования параметров. В форме редактирования в основных параметрах потока укажите для параметра «Group» значение из доступных в выпадающем списке. Поток будет привязан к выбранной группе. Нажмите кнопку «Save» в правом верхнем углу формы.

9.3 Вкладка «Queue»

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 свободный «рабочий процесс FaceStream» потока с запросом на пул из очереди, поток принимается к обработке и ему присваивается статус «in_progress».

См. подробную информацию в разделе [«Процесс обработки потоков в LUNA Streams»](#).

Вкладка «Queue» носит информационный характер и предназначена для отображения всех потоков, находящихся в очереди на обработку, их идентификаторов и статусов.



Queue		
Stream ID	Stream name	Status
b5254e26-4f47-4c99-a676-b133f2f0cebb	Test name 2	pending
af3b977d-40d4-4b60-99b9-8274758d0cb7	Test name	pending
		25 ▾

Рис. 27: Вкладка «Queue»

Вкладка «Queue» содержит следующие элементы и параметры потока:

- перечень потоков:
 - «Stream ID» — идентификатор видеопотока в LUNA Streams, генерируется при создании потока;
 - «Stream name» — название видеопотока;
 - «Status» — текущий статус видеопотока;
- количество отображаемых на странице потоков задается переключателем в нижнем правом углу страницы. Всего может быть 10, 25, 50 или 100 потоков на одной странице.

10 Мониторинг

Мониторинг в FaceStream реализован как [отправка данных в InfluxDB](#) и отключен по умолчанию.

В LUNA Streams есть несколько методов мониторинга:

- [Отправка данных в InfluxDB](#) (включен по умолчанию)
- [Экспорт метрик в формате Prometheus](#) через ресурс `/metrics` (отключен по умолчанию)

10.1 InfluxDB

Для работы с InfluxDB необходимо зарегистрироваться с помощью логина и пароля и указать имя бакета, имя организации и токен. Все эти данные задаются при запуске контейнера InfluxDB с помощью переменных окружения.

Для того, чтобы использовать мониторинг FaceStream или LUNA Streams, необходимо в [настройках FaceStream](#) или [настройках LUNA Streams](#) задать для полей «bucket», «organization», «token» точно такие же данные, которые указывались при запуске контейнера InfluxDB. Так, например, если при запуске контейнера InfluxDB использовались следующие настройки...

```
-e DOCKER_INFLUXDB_INIT_BUCKET=luna_monitoring \  
-e DOCKER_INFLUXDB_INIT_USERNAME=luna \  
-e DOCKER_INFLUXDB_INIT_PASSWORD=password \  
-e DOCKER_INFLUXDB_INIT_ORG=luna \  
-e DOCKER_INFLUXDB_INIT_ADMIN_TOKEN=kofqt4Pfqn6o \
```

... то в настройках FaceStream или LUNA Streams должны быть указаны следующие параметры:

```
"influxdb": {  
  "organization": "luna",  
  "token": "kofqt4Pfqn6o",  
  "bucket": "luna_monitoring",
```

Логин и пароль используются для доступа к пользовательскому интерфейсу InfluxDB.

Настройки FaceStream и LUNA Streams содержат разные данные полей «bucket», «organization» и «token» по умолчанию. Если необходимо использовать мониторинг для обоих сервисов, то нужно задать одинаковые настройки. При необходимости можно сохранять данные FaceStream и LUNA Streams в разные бакеты (см. ниже).

Для того, чтобы разделить данные мониторинга FaceStream и LUNA Streams, можно создать отдельные бакеты после запуска контейнера InfluxDB. Это можно сделать с помощью одного из следующих способов:

- с помощью пользовательского интерфейса InfluxDB (вкладка Explore > Create bucket) после старта контейнера InfluxDB
- с помощью команды `influx bucket create -n <bucket_name> -o <organization_name>` в InfluxCLI после старта контейнера InfluxDB

Обратите внимание, что имя организации должно быть таким же, как при создании контейнера InfluxDB.

Отправляемые данные в InfluxDB отличаются для LUNA Streams и FaceStream. См. соответствующие разделы ниже для более подробной информации об отправляемых данных.

10.2 Мониторинг FaceStream

10.2.1 Включение мониторинга

Для включения мониторинга FaceStream нужно выполнить следующие действия:

- перейти в пользовательский интерфейс Configurator: `http://<configurator_server_ip>:5070/`
- ввести «FACE_STREAM_CONFIG» в поле «Setting name» и нажать «Apply Filters»
- включить настройку «[send_data](#)» в секции «[monitoring](#)»
- в зависимости от значений параметров «DOCKER_INFLUXDB_INIT_BUCKET», «DOCKER_INFLUXDB_INIT_ORG», «DOCKER_INFLUXDB_INIT_ADMIN_TOKEN» задаваемых при запуске контейнера InfluxDB, указать соответствующие значения в поля «bucket», «organization» и «token» в секции «[monitoring](#)».
- перезапустить контейнер FaceStream: `docker restart facestream`

10.2.2 Отправляемые данные

В InfluxDB отправляются следующие элементы:

- элемент `measurements`. Он равен значению `fs-requests`.
- набор тегов:
 - `fs_ip` — IP-адрес, где развернут FaceStream
 - `source` — поле «name», задаваемое при создании потока в LUNA Streams (опционально)
 - `stream_id` — идентификатор потока
- набор полей:
 - `track_id` — идентификатор трека
 - `event_id` — идентификатор события, полученный от LUNA PLATFORM

- request_id — внешний идентификатор для связи с мониторингом сервисов LUNA PLATFORM
- track_start_time — время начала трека
- track_best_shot_time — время, когда кадр с отправляемым лучшим снимком появился в системе
- track_best_shot_min_size_time (опционально) — время, когда размер детекции достиг значения, заданного в параметре «best_shot_min_size»
- track_best_shot_proper_size_time (опционально) — время, когда размер детекции достиг значения, заданного в параметре «best_shot_proper_size»
- liveness_start_time (опционально) — время начала работы Liveness
- liveness_end_time (опционально) — время окончания работы Liveness
- bestshot_count — количество отправленных лучших снимков в одном запросе в LP вместе с текущим лучшим снимком. Так, например, если было выполнено 2 отправки по 10 лучших снимков, то значение данного параметра будет равно 10, а значение параметра track_send_count равно 2
- time_from_first_frame_to_send — время, которое прошло от появления первого кадра в FS до отправки в LP
- track_send_count — порядковый номер отправки данных с трека

Поля, содержащие время, отправляются в виде UTC с точностью до микросекунд.

- элемент timestamp. Он равен времени отправки лучшего снимка или лучших снимков в микросекундах.

Частота отправки данных в InfluxDB регулируется параметром «flashing_period» настроек FaceStream.

Лучших снимков может быть несколько, т.к. отправка с одного трека одновременно считается одним измерением. Для сохранения этого измерения в InfluxDB используются данные последнего лучшего снимка из группы лучших снимков. Данные, которые уникальны для каждого лучшего снимка (track_best_shot_time, liveness_start_time, liveness_end_time) будут теряться для всех лучших снимков кроме последнего в случае такой отправки.

Если опциональные поля отсутствуют, то данные этих полей не будут отправлены в БД Influx.

При нормальной работе мониторинга, в логи FaceStream не выводится дополнительной информации. Если при мониторинге выявлена ошибка, то соответствующее сообщение появится в логах FaceStream.

10.3 Мониторинг LUNA Streams

10.3.1 Отправляемые данные в InfluxDB

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

Каждое событие — это точка временного ряда. Для сервиса API точка представлена с использованием следующих данных:

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

Для других сервисов набор типов событий может отличаться. Например, сервис Handlers также собирает данные об использовании SDK, выполненных оценках и лицензировании.

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.

Сохранение данных для серии запросов запускается при каждом запросе. Каждая точка содержит данные о соответствующем запросе (время выполнения и т.д.).

- теги

Имя тега	Описание
service	сервис, всегда «luna-streams»
route	объединение метода запроса и ресурса запроса (POST:/streams)
status_code	HTTP статус код ответа

- поля

Имя поля	Описание
request_id	идентификатор запроса
execution_time	время выполнения запроса

Сохранение данных для серии ошибок запускается при сбое запроса. Каждая точка содержит *error_code* ошибки LUNA.

- теги

Имя тега	Описание
service	сервис, всегда «luna-streams»
route	объединение метода запроса и ресурса запроса (POST:/streams)
status_code	HTTP статус код ответа
error_code	код ошибки LUNA PLATFORM

- поля

Имя поля	Описание
request_id	идентификатор запроса

Сохранение данных для серии лицензирования запускается при старте сервиса и каждые 60 секунд. Каждая точка содержит данные проверки лицензии.

- теги

Имя тега	Описание
service	сервис, всегда «luna-streams»
license_status	статус лицензии («ok», «warning», «error», «exception»)

- поля

Имя поля	Описание
license_streams_limit_rate	процент использованных запросов
warnings	предупреждающие сообщения о лицензии
errors	сообщения об ошибках лицензии

10.3.2 Просмотр данных InfluxDB

Для просмотра данных мониторинга можно использовать интерфейс InfluxDB.

- перейдите в пользовательский интерфейс InfluxDB `<server_ip>:<influx_port>`. Порт по умолчанию — 8086. Данные для входа по умолчанию — luna/password.
- выберите вкладку Explore
- выберите способ отображения информации в выпадающем списке (график, гистограмма, таблица и пр.)
- выберите бакет внизу страницы
- отфильтруйте необходимые данные
- нажмите «Submit».

10.3.3 Экспорт метрик в формате Prometheus

Сервис LUNA Streams может собирать и сохранять метрики в формате Prometheus в виде данных временных рядов, которые можно использовать для отслеживания поведения сервиса. Метрики могут быть интегрированы в систему мониторинга Prometheus для отслеживания производительности. См. [официальную документацию Prometheus](#) для подробной информации.

По умолчанию сбор метрик отключен. Сбор метрик включается в группе параметров «LUNA_SERVICE_METRICS».

Обратите внимание, что все данные метрик сбрасываются при завершении работы сервиса.

10.3.3.1 Тип метрик

Доступно два типа метрик:

- **счетчики**, которые увеличиваются с каждым событием
- **кумулятивные гистограммы**, которые используются для измерения распределения продолжительности или размера событий

Кумулятивная гистограмма — это отображение, которое подсчитывает совокупное количество наблюдений во всех интервалах до указанного интервала. См. описание в [Wikipedia](#).

Доступны следующие метрики типа **счетчики**:

- `request_count_total` — общее количество запросов
- `errors_count_total` — общее количество ошибок

Каждый из них имеет как минимум два лейбла для сортировки:

- `status_code` (или `error_code` для метрик ошибок)
- `path` — путь, состоящий из метода запроса и маршрута конечной точки

Лейблы представляют собой ключевые пары, состоящие из имени и значения, которые присваиваются метрикам.

При необходимости можно добавить пользовательские типы лейблов, указав пару `label_name=label_value` в параметре «`extra_labels`».

Обратите внимание, что пара `label_name=label_value` будет добавлена к каждому сервису LUNA PLATFORM.

Специальный менеджер распределяет все запросы, проходящие через сервис, среди счетчиков, используя эти метки. Это обеспечивает, что два успешных запроса, отправленных на разные конечные точки или на одну конечную точку, но с разными кодами состояния, будут доставлены к различным метрикам.

Неудачные запросы распределяются по метрикам `request_count_total` и `request_errors_total`.

Метрика `requests` типа **кумулятивные гистограммы** отслеживает длительность запросов к сервису. Для гистограммы определяются следующие интервалы (бакеты), в которые попадают измерения:

- 0.0001
- 0.00025
- 0.0005
- 0.001
- 0.0025
- 0.005
- 0.01
- 0.025
- 0.05
- 0.075
- 0.1
- 0.25
- 0.5
- 0.75
- 1.0
- 2.5
- 5.0
- 7.5
- 10.0
- Inf

Таким образом диапазон времени запроса может быть разбит на несколько интервалов, начиная от очень быстрых запросов (0.0001 секунды) до очень долгих (Inf - бесконечность). У гистограмм

также есть метки для классификации данных, такие как `status_code` для статуса запроса или `route` для указания маршрута запроса.

Примеры

Если отправить один запрос в ресурс `/healthcheck`, за которым последуют три запроса в ресурс `/docs/сpec`, один из которых будет перенаправлен (код состояния ответа 301), то при выполнении запроса к ресурсу `/metrics`, в теле ответа будет выдан следующий результат:

```
# HELP request_count_total Counter of requests
# TYPE request_count_total counter
request_count_total{path="GET:/docs/spec",status_code="200"} 2.0
request_count_total{path="GET:/docs/spec",status_code="301"} 1.0
request_count_total{path="GET:/healthcheck",status_code="200"} 1.0
```

Если отправить один неверный POST-запрос к ресурсу `/streams`, то при выполнении запроса к ресурсу `/metrics`, в теле ответа будет выдан следующий результат:

```
# HELP request_count_total Counter of requests
# TYPE request_count_total counter
request_count_total{path="POST:/streams",status_code="401"} 1.0
# HELP request_errors_total Counter of request errors
# TYPE request_errors_total counter
request_errors_total{error_code="12010",path="POST:/streams"} 1.0
# HELP requests Histogram of request time metrics
# TYPE requests histogram
requests_sum{route="GET:/docs/spec",status_code="200"} 0.003174567842297907
requests_bucket{le="0.0001",route="GET:/docs/spec",status_code="200"} 0.0
requests_bucket{le="0.00025",route="GET:/docs/spec",status_code="200"} 0.0
requests_bucket{le="0.0005",route="GET:/docs/spec",status_code="200"} 0.0
requests_bucket{le="0.001",route="GET:/docs/spec",status_code="200"} 1.0
...
requests_count{route="GET:/docs/spec",status_code="200"} 2.0
requests_sum{route="GET:/docs/spec",status_code="301"} 0.002381476051209132
```

10.3.4 Настройка сбора метрик для Prometheus

Prometheus необходимо настроить на сбор метрик LUNA PLATFORM.

Пример [конфигурации](#) Prometheus для сбора метрик сервисов LP:

```
- job_name: "luna-streams"
  static_configs:
```

```
- targets: ["127.0.0.1:5160"]
...

- job_name: "luna-configurator"
  static_configs:
    - targets: ["127.0.0.1:5070"]
```

См. пример запуска Prometheus в [официальной документации](#).

11 Вывод информации в логи

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

11.1 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.

11.2 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.

11.2.1 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».

11.2.2 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 введён некорректный статус в поле «status». Можно задавать только два статуса — «pause» и «pending». Остальные статусы можно получить только в определённый момент времени (см. раздел [«Таблица переходов от статуса к статусу»](#)). В описании ошибки приводится ожидаемый статус.

11.2.3 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).

11.2.4 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.

11.2.5 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.

11.2.6 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».

11.2.7 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».

11.2.8 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».

11.2.9 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.

11.2.10 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.

12 Дополнительная информация

12.1 Диаграмма активности для Liveness и Primary Track Policy

Ниже приведена диаграмма активности работы FaceStream при использовании Liveness и политики Главного Трека.

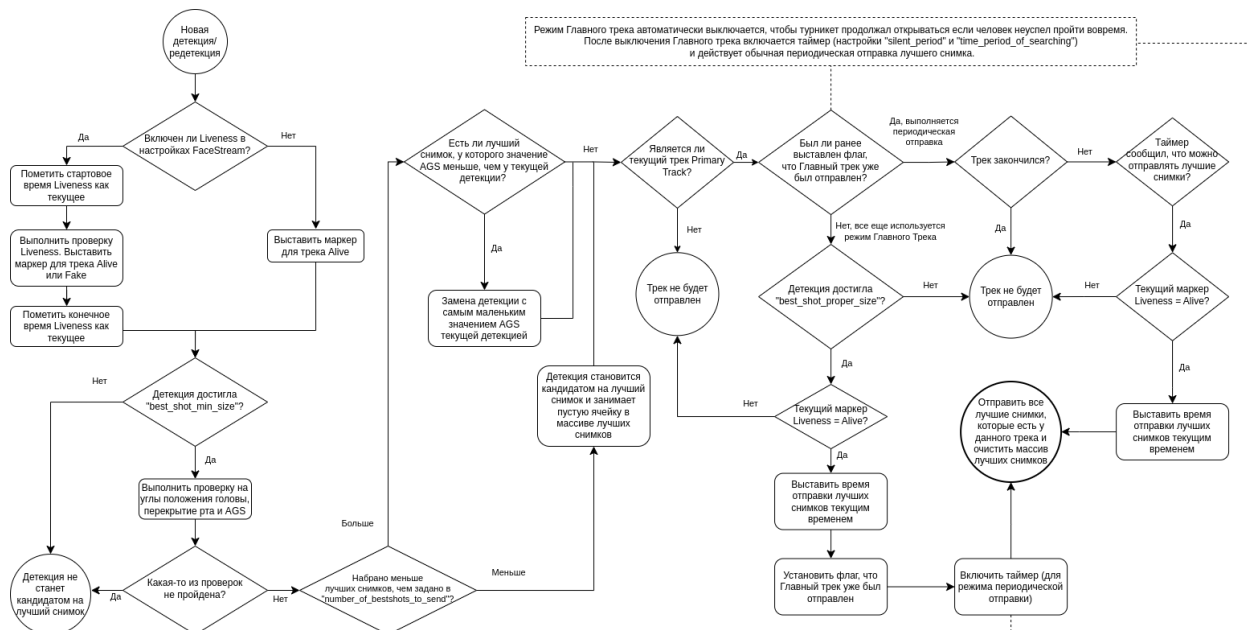


Рис. 28: Диаграмма активности для Liveness и Primary Track Policy

12.2 Особенности работы с предварительным просмотром потока

Для того, чтобы записывать в поток LUNA Streams адрес предварительного просмотра потока (поле `preview > live > url`), FaceStream должен определять свой IP-адрес. Он определяет свой IP-адрес одним из двух способов:

- через подключение к сервису LUNA Streams
- через переменную окружения системы `VL_FACE_HOST`. Переменная окружения может быть задана с помощью выполнения следующей команды в терминале сервера: `export VL_FACE_HOST=<your_ip_address>`.

Если IP-адрес переменной `VL_FACE_HOST` задан неверно, то LUNA Streams будет удалять потоки с некорректным адресом, а FaceStream завершит работу следующей ошибкой `Failed to validate input json`.

Использование переменной окружения переопределяет способ подключения к сервису LUNA Streams.

Если IP-адрес FaceStream определяется путем подключения к сервису LUNA Streams, то для определения своего IP-адреса, FaceStream всегда должен запускаться после LUNA Streams. Если по каким-то причинам FaceStream запустился перед LUNA Streams (например, выполнен перезапуск сервера, где запущены и FaceStream и LUNA Streams), то FaceStream может завершить работу ошибкой `Failed to get local IP address. Reason`. Если IP-адрес FaceStream определяется через переменную окружения системы `VL_FACE_HOST`, то вышеописанная ошибка не возникнет.

12.3 Проксирование запросов в LUNA Streams через LUNA API

Если FaceStream используется совместно с LUNA PLATFORM, то доступна возможность отправлять часть запросов в сервис LUNA Streams через сервис LUNA API.

Для этого необходимо включить и настроить специальный встроенный плагин «luna-streams.py» в сервисе API. Плагин расположен в контейнере сервиса API по пути `/srv/luna_api/plugins` и включается с помощью настройки «LUNA_API_ACTIVE_PLUGINS».

См. более подробную информацию о механизме плагинов в разделе «Плагины» руководства администратора LUNA PLATFORM.

Для плагина доступны следующие настройки, указываемые в конфигурации «LUNA_API_PLUGINS_SETTINGS» сервиса API:

- «luna-streams-address» > «origin» — протокол, IP адрес и порт сервиса LUNA Streams
- «luna-streams-address» > «api_version» — версия API сервиса LUNA Streams
- «luna-streams-timeouts» > «connect» — таймаут для установки соединения при отправке HTTP-запроса к сервису LUNA Streams
- «luna-streams-timeouts» > «request» — общий таймаут для выполнения всего HTTP-запроса
- «luna-streams-timeouts» > «sock_connect» — таймаут для установки соединения на уровне сокетов
- «luna-streams-timeouts» > «sock_read» — таймаут на чтение данных с сокета после успешного соединения

При отсутствии указания конфигурации «LUNA_API_PLUGINS_SETTINGS» или вышеописанных настроек, будут применены значения по умолчанию. См. подробную информацию о вышеперечисленных настройках и их значениях по умолчанию в разделе «LUNA_API_PLUGINS_SETTINGS» руководства администратора LUNA PLATFORM.

Пример задания настройки «LUNA_API_PLUGINS_SETTINGS»:

```
{
  "luna-streams": {
    "luna-streams-address": {
      "origin": "http://127.0.0.1:5160",
      "api_version": 2
    },
    "luna-streams-timeouts": {
      "request": 60,
      "connect": 20,
      "sock_connect": 10,
      "sock_read": 60
    }
  }
}
```

```
}
```

При необходимости можно разграничить права доступа для запросов к LUNA Streams с помощью разрешений для токена. Для этого нужно создать новый или обновить существующий токен, передав пользовательский ключ «streams» с разрешениями для токена.

Для проксирования запросов из сервиса LUNA API к сервису LUNA Streams нужно указывать разрешение с именем «streams». Другие пользовательские названия ключей не подходят.

Можно задать следующие разрешения:

- «creation» (POST-запросы)
- «view» (GET-запросы)
- «modification» (PUT-запросы и PATCH-запросы)
- «deletion» (DELETE-запросы)

Пример указания пользовательского ключа «streams» со всеми возможными разрешениями в теле запроса на создание токена:

```
{
  "permissions": {
    "streams": [
      "creation",
      "view",
      "modification",
      "deletion"
    ]
  }
}
```

См. подробную информацию о токенах в разделе «Аккаунты, токены и способы авторизации» руководства администратора LUNA PLATFORM.

В таблице ниже отражено соотношение маршрутов в LUNA API и LUNA Streams, а также ссылки на запросы и возможные разрешения токена.

Маршрут в LUNA API	Маршрут в LUNA Streams	Запросы	Разрешение токена
prefix/version	/version	get version	Нет
prefix/docs/spec	{api_version}/docs/spec	get openapi documentation	Нет
prefix/plugins	{api_version}/plugins	get list of plugins	Нет

Маршрут в LUNA API	Маршрут в LUNA Streams	Запросы	Разрешение токена
prefix/streams	{api_version}/streams	create stream	Пользовательский
		get streams	
		delete streams	
prefix/streams/count	{api_version}/streams/count	delete streams	Пользовательский
prefix/streams/{stream_id}	{api_version}/streams/{stream_id}	get stream	Пользовательский
		update stream	
		put stream	
		remove stream	
prefix/streams/logs	{api_version}/streams/logs	get streams logs	Пользовательский
		delete streams logs	
prefix/streams/{stream_id}/preview/handler/live	{api_version}/streams/{stream_id}/preview/handler/live	get live preview	Пользовательский
prefix/streams/{stream_id}/preview/handler/frame	{api_version}/streams/{stream_id}/preview/handler/frame	get last frame preview	Пользовательский

prefix — /6/plugins/luna-streams.

Можно отправлять запросы как в первую версию API LUNA Streams, так и во вторую. Для этого нужно использовать заголовок LUNA-STREAMS-API-VERSION следующим образом:

- если заголовок LUNA-STREAMS-API-VERSION не указан, будет автоматически выбрана первая версия API
- если значение заголовка LUNA-STREAMS-API-VERSION равно «1», будет выбрана первая версия
- если значение заголовка LUNA-STREAMS-API-VERSION равно «2», будет выбрана вторая версия API

- в противном случае вернется ошибка

12.4 Данные событий, генерируемых с помощью FaceStream

Обрабатывая потоки, FaceStream передает запросы на генерацию событий в LUNA PLATFORM. Для генерации события необходимо при создании потока заполнить поле «event_handler», указывая идентификатор обработчика в поле «bestshot_handler», адрес и версию API сервиса LUNA API и адрес сохранения исходного кадра (при необходимости).

Ниже приведена таблица, поясняющая какие данные, обработанные FaceStream, записываются в событие LUNA PLATFORM.

Имя параметра	Описание
account_id	Идентификатор аккаунта, задаваемый при создании потока в LUNA Streams.
create_time	Время начала трека (время возникновения события в видеопотоке) относительно времени сервера FS.
end_time	Время окончания трека (время окончания события в видеопотоке) относительно времени сервера FS.
handler_id	Идентификатор обработчика, задаваемый при создании потока в LUNA Streams.
stream_id	Идентификатор потока в LUNA Streams.
source	Поле «name», задаваемое при создании потока в LUNA Streams.
track_id	Номер трека FaceStream.
face_detections:	Данные об обнаружении лица: * sample_id : Идентификатор БО (лучшего снимка), создаваемый если включена политика «face_sample_policy». * detect_time : Время обнаружения лучшего снимка лица относительно времени сервера FS. * image_origin : Исходный кадр. * detect_ts : Время обнаружения лучшего снимка лица относительно начала видеофайла. * detection : Координаты ограничивающего треугольника с детекцией лица («x», «y», ширина, высота).
body_detections:	Данные об обнаружении тела: * sample_id : Идентификатор БО (лучшего снимка), создаваемый если включена политика «body_sample_policy». * detect_time : Время обнаружения лучшего снимка тела относительно времени сервера FS.

Имя параметра	Описание
	* image_origin : Исходный кадр.
	* detect_ts : Время обнаружения лучшего снимка тела относительно начала видеофайла.
	* detection : Координаты ограничивающего треугольника с детекцией тела («x», «y», ширина, высота).
location	Данные о местоположении, задаваемые при создании потока в LUNA Streams.

Кроме вышеописанных параметров, событие будет дополнено параметрами, генерируемыми по указанному обработчику (например, «face_id», «match_result», «attach_result», «gender», и др.). Поля «external_id» и «user_data» всегда будут оставаться пустыми при генерации события по биометрическому образцу, передаваемому из FaceStream.