



VisionLabs
MACHINES CAN SEE

VisionLabs FaceStream

Release notes

Contents

FaceStream v.5.84.0	4
FaceStream v.5.81.2	4
FaceStream v.5.2.3	5
FaceStream v.5.2.2	5
FaceStream v.5.2.0	6
FaceStream v.5.2.0	7
FaceStream v.5.1.49	9
FaceStream v.5.1.46	10
FaceStream v.5.1.45	10
FaceStream v.5.1.43	11
FaceStream v.5.1.41	12
FaceStream v.5.1.32	14
FaceStream v.5.1.30	14
FaceStream v.5.1.28	15
FaceStream v.5.1.27	15
FaceStream v.5.1.23	17
FaceStream v.5.1.20	18
FaceStream v.5.1.18	19
FaceStream v.5.1.16	21
FaceStream v.5.1.14	22
FaceStream v.5.1.13	23
FaceStream v.5.1.12	24
FaceStream v.5.1.10	24

FaceStream v.5.1.9	25
FaceStream v.5.1.8	26
FaceStream v.5.1.7	27
FaceStream v.5.1.6	28
FaceStream v.5.1.5	28
FaceStream v.5.1.4	29
FaceStream v.5.1.3	30
FaceStream v.5.1.2	31
FaceStream v.5.1.1	32
FaceStream v.5.1.0	32
FaceStream v.5.0.7	35
FaceStream v.5.0.6	35

FaceStream v.5.84.0

Fixed errors

- Fixed an error where the “fps_floor” parameter (frame skip control) did not work properly when using variable frame rate video streams.

Frame rate in such videos is now calculated in real time using the average difference between the timestamps (PTS) of adjacent frames.

FaceStream v.5.81.2

Changes

- FaceStream is now part of the LUNA PLATFORM distribution.

The FaceStream distribution now has a LUNA PLATFORM version to avoid component compatibility issues. In other words, now FaceStream of a certain version always works with LUNA PLATFORM of the same version.

The documentation on installing the FaceStream Standalone version has been removed from the distribution and is now available upon request.

Important: The FaceStream container still has its own version. It is the version of the FaceStream component distribution that has changed. For example, for the FaceStream v.5.82.1 distribution, the FaceStream component (container) version is v.5.2.6.

- FSDK has been updated to version v.5.22.1.

Fixed errors

- Fixed an error that prevented FaceStream from running from a Docker container with configuration files.

Now a new environment variable “CONFIGURATION_PATH” has been added to the FaceStream container, defining the path to the configuration file “fs3Config.conf”.

By default, the configuration file at “/srv/facestream/data/fs3Config.conf” is used.

The configuration file is used only if the environment variables “CONFIGURATOR_HOST” and “CONFIGURATOR_PORT” are not passed, which are necessary to use the settings from the Configurator service.

Thus, the command to run FaceStream with a configuration file can look like this:

```
docker run \  
--env=CONFIGURATION_PATH=/srv/facestream/data/fs3Config.conf \  
facestream
```

```
-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/runtime.conf:/srv/facestream/data/runtime.conf \
-v /etc/localtime:/etc/localtime:ro \
-v /tmp/logs/facestream:/srv/logs \
--env=PORT=34569 \
--detach=true \
--restart=always \
--name=facestream \
--network=host \
dockerhub.visionlabs.ru/luna/facestream:v.5.2.6
```

- Fixed excessively long execution of “[get last frame preview](#)” and “[get live preview](#)” requests with a large number of log records in the Streams DB.

FaceStream v.5.2.3

Changes

- A new parameter “fps_floor” has been added to the FaceStream settings, which is intended to control the number of analyzed frames in the video stream, allowing for reduced system load by skipping some frames. This can be useful when it is necessary to lower computational resource requirements while maintaining acceptable analysis accuracy.

The number of skipped frames is calculated as the difference between the video stream’s frame rate and the “fps_floor” value. Skipped frames are evenly distributed over each second.

If the “fps_floor” value is zero or negative, FaceStream will ignore the setting, and the number of processed frames will remain unchanged.

For example, with “fps_floor” set to 15 for a video stream with a frame rate of 25 fps, FaceStream will skip 10 frames per second (25 - 15). These 10 frames will be evenly distributed over time to maintain video playback integrity and avoid sudden jumps.

- The request to the “[/version](#)” resource in FaceStream now returns the FSDK version in use.

FaceStream v.5.2.2

Changes

- The “INFLUX_MONITORING” section has been renamed to “LUNA_MONITORING” in the LUNA Streams settings.

The section now also includes the “storage_type” parameter, which defines the type of monitoring data storage. Currently, only the “influx” type is available.

Important: Since a similar update has been added to LUNA PLATFORM v.5.76.0, the minimum required version of LUNA PLATFORM for FaceStream has been updated to v.5.76.0.

- Now, if detection does not reach the value specified in the “[min_body_size_threshold](#)” parameter, it will not be considered the best shot.
- The [automatic stream restart mechanism](#) has been improved for more efficient processing of large data volumes.

Fixed errors

- The ability to use the “[preferred_program_stream_frame_width](#)” parameter in the LUNA Streams API V2 has been restored.
- Fixed errors that occurred when performing background tasks for automatic stream restarts with a large number of streams.
- In the “[create stream V2](#)” request, at least one element in the “data” > “analytics” section must now be specified.
- Fixed various errors that occurred when sending reports during a large number of concurrent requests with many streams.

FaceStream v.5.2.0

Changes

- The section “INFLUX_MONITORING” has been renamed to “LUNA_MONITORING” in LUNA Streams settings.

The section has also been updated with the “storage_type” parameter, which defines the type of monitoring data storage. Currently, only the “influx” type is available.

Important: Since a similar update has been added to LUNA PLATFORM v.5.76.0, the minimum required version of LUNA PLATFORM for working with FaceStream has been updated to v.5.76.0.

- Detection will no longer be considered the best shot if it does not reach the value specified in the “[min_body_size_threshold](#)” parameter.
- The mechanism for [automatic stream restart](#) has been improved for more effective processing of large data volumes.

Fixed errors

- The missing parameter [“preferred_program_stream_frame_width”](#) has been returned in the API V2 of LUNA Streams service.
- Errors occurring during background tasks of automatic stream restart with a large number of streams have been fixed.
- In the [“create stream V2”](#) request, it is now required to specify at least one item in the “data” > “analytics” section.
- Various errors occurring when sending reports during a high volume of simultaneous requests with many streams have been fixed.

FaceStream v.5.2.0

Changes

- Support for API V2 of the LUNA Streams service has been added.

The changes in the second version are aimed at using the collaborative mode of operation of the face and body detector. The face and body detector is enabled by simultaneously enabling the “use-face-detector” and “use-body-detector” parameters in the TrackEngine settings.

API V1 support remains available.

Main changes in the new version of the API:

- Setting the address and API version of the LUNA API service has been removed. Now the connection settings with LUNA PLATFORM are specified in the FaceStream settings in the “lunaplatform” section.
- The “min_score” parameter is divided into two parameters - “min_score_face” and “min_score_body”.
- Parameter “number_of_bestshots_to_send” is divided into two parameters - “face_bestshots_to_send” and “body_bestshots_to_send”.
- Removed all deprecated parameters (for example, “detection_handler”).

The administrator manual has been completely redesigned to use API V2.

The plugin for proxying LUNA Streams requests via the LUNA API service also supports the new version of the API. See [“Proxying requests to LUNA Streams using LUNA API”](#) for details.

The LUNA Streams user interface does not currently support API V2. Support will be added later.

Important: At the moment there are restrictions on the operation of the API V2 functionality described below.

General functionality:

- Missing parameter “preferred_program_stream_frame_width”.

- It is impossible to specify more than one analytics in an analytics array.

Collaborative mode of operation of faces and bodies:

- Section “data” > “analytics” > “primary_track_policy” is not supported.
- Parameter “data” > “analytics” > “sending” > “full_frame_settings”, which determines which source frames will be sent (faces, bodies or faces and bodies), is not supported. Currently, only source frames of faces and bodies are sent.
- Parameter “data” > “analytics” > “sending” > “bestshot_settings” > “type”, which determines which best shots will be sent (faces, bodies or faces and bodies), is not supported. At this time, only the best faces and bodies will be submitted.

Periodic sending of best photos in shared mode works similarly to working with bodies. This means that the “silent_period” parameter will not work in shared mode.

A description of the limitations of the collaborative mode is also described in the section [“Face and body detection collaborative mode”](#) in the administrator manual.

- The ability to send Crop images to an external service instead of samples (“warp” format) and modified images (“gost” format) has been added.

Sending is controlled by the “sending” > “portrait_type” parameter, which now accepts the new value “crop”.

A Crop image is a fragment of an image obtained by expanding and cropping the area around the detected object. The expansion of the area is determined by the new “sending” > “crop_factor” parameter, which increases the dimensions of the original rectangle containing the object. The maximum size of the cropped image is controlled by the new “sending” > “max_crop_size” parameter, which limits the largest side of the area. If the size of the fragment exceeds the specified parameter, the fragment will be scaled proportionally along both coordinates to the desired size. A Crop image allows capturing more context and detail around the object compared to a sample.

This allows for more flexible image processing. Sending Crop images has several key advantages:

1. **Face coordinates:** Along with the Crop image, the coordinates of the face on both the Crop image and the source image are transmitted. This avoids the need to use redetector, ensuring the presence of the face in the image and saving computational resources.
2. **Additional information:** A Crop image contains more context and details around the face, allowing for the estimation of various parameters that cannot be estimated on a biometric. For example, Deepfake estimation, Liveness estimation, etc., can be performed with greater accuracy.
3. **External services:** External services and face recognition algorithms often work better with Crop images than with samples. This is because Crop images contain more information and context, improving the accuracy of recognition and analysis.

Important: Starting from the current version of FaceStream, it is necessary to use LUNA PLATFORM v.5.67.0 and higher. Older versions of LUNA PLATFORM are not supported by the current version of FaceStream.

FaceStream v.5.1.49

Changes

- Now all human face data, including face samples, is sent to the LUNA PLATFORM in one request to the `"/6/handlers/{handler_id}/stream_events"` resource.

If the `"frame_store"` parameter of the stream management settings specifies the address to the `"/images"` resource and the `"send_source_frame"` parameter is enabled in the FaceStream settings, then the source images will also be attached to the request.

The structure of the returned event remains the same.

The track and request IDs are recorded in the FaceStream logs. This enables you to track the sending of a request to LUNA PLATFORM.

Example logs:

```
[17:28:53.684796 4175687 LunaBaseClient.cpp:98] [client] Generated
request to LUNA (API 6). Luna-Request-Id = 1712327333,d880445b-6dde
-4da8-bd1b-88d38cce7814, Luna-Event-Time = 2024-04-05T14
:28:47.740913Z, Luna-Account-Id = b48efe64-9b3c-4be9-8300-5
fd1469dde7b. Track = 63f1f93e-152a-47ec-bc31-0eff0000000a
identification = Source 0
[I0405 17:28:53.684837 4175687 LunaBaseClient.cpp:60] [client] Request
[I0405 17:28:53.733474 4175722 LunaClient.cpp:31] parse response on
creation of a new event.
[I0405 17:28:53.734727 4175722 StreamEventsSender.cpp:172] [client] Got
successful response for request (id 1712327333,ee37cb41-be35-4dda-9
f70-e0633d3dc79c). Status code: 201
```

The first line of the log contains the request and track IDs, and the last line contains the ID and status of the request from LUNA PLATFORM.

- In the container of the LUNA Streams service, the Python version has been updated to 3.12.

Support for older versions of Python has been discontinued.

- Performance of the `"get streams"` request has been improved in some cases.
- The system requirements for FaceStream have been moved to a separate document in the distribution package.

Fixed errors

- Fixed the error in the LUNA Streams service that caused an error in clearing stream logs when the database was unavailable.
- Fixed FaceStream crashing when the “location” field in the stream control settings was not populated.
- Fixed issue with processing incorrect request to InfluxDB in some cases.
- Fixed incorrect FaceStream operation on A-series, 30xx video cards with “frg-subtractor” parameter enabled.

FaceStream v.5.1.46

Changes

- When using ROI, FaceStream now sends the uncropped source frame to LUNA PLATFORM.
Note: The size of the source frame sent is also increased.
Also added to the preview page is the ROI display as an orange bounding box.
- Video memory consumption when calculating Mask Liveness on the GPU has been reduced.

Fixed errors

- Fixed the error where the bounding boxes on the preview page were the wrong color and the frame counter was missing.
See section “[Stream playback interface](#)” for information about the colors of the bounding boxes on preview page.
- The FaceStream message about the need to read an additional frame has been moved to the WARNING category, since it is not an error.
- Fixed the error in the LUNA Streams service that did not allow creating new stream log entries in the Streams database due to its overflow.

FaceStream v.5.1.45

Changes

- FSDK has been updated to version 5.16.1.
- The request parameters order, stream_id__gte and stream_id__lt have been added to the [get streams](#) request.

Fixed errors

- Detections will now not be sent to LUNA PLATFORM if the “minimal_body_track_length_to_send” parameter is set to “0”.

However, the list of detections will not be empty, because the best shots will be sent as detections when the “number_of_bestshots_to_send” parameter is set to a non-zero value.

FaceStream v.5.1.43

Changes

- Now all human body data, including body samples, is sent to the LUNA PLATFORM in one request to the “/6/handlers/{handler_id}/stream_events” resource.

Previously, human body data was sent in two requests, which generated two events:

- General event generated by the static handler, containing all the basic data about the human bodies, including the address of the stored body sample.
- Auxiliary event generated by the dynamic handler, containing only the coordinates of the human body, basic information about the auxiliary event (detection time, event creation time, account ID, etc.) and a “track_id” for connection with the general event.

Now that everything is sent to a single resource, you no longer need to create a dynamic handler and specify it in the “detection_handler” > “handler_id” parameter. Specifying such a handler will be ignored. In the LUNA Streams OpenAPI specification, the “detection_handler” field is marked as **Deprecated**.

If the “frame_store” parameter of the stream management settings specifies the address to the “/images” resource and the “send_source_frame” parameter is enabled in the FaceStream settings, then the source images will also be attached to the request.

The structure of the returned event remains the same.

Important: Starting from the current version of FaceStream, it is necessary to use LUNA PLATFORM v.5.56.0 and higher. Older versions of LUNA PLATFORM are not supported by the current version of FaceStream.

- A built-in plugin has been added to the LUNA API v.6.22.0 service (LUNA PLATFORM v.5.56.0 version), which enables you to perform some requests to the LUNA Streams service through the API service.

See [release notes for LUNA PLATFORM v.5.56.0](#) and the “Proxying requests to LUNA Streams using LUNA API” section in the FaceStream administrator manual for more details.

- The calculation of Mask Liveness has been accelerated and memory consumption has been reduced during the calculation.
- A new request “[get list of plugins](#)” has been added to the LUNA Streams service, which enables you to get a list of imported plugins and their status.

Fixed errors

- Fixed a memory leak that occurred in some cases when the preview was turned on. Reduced RAM consumption when this option is enabled.

FaceStream v.5.1.41

- Starting from the current version of FaceStream, it is necessary to use LUNA PLATFORM v.5.53.0 and higher. Older versions of LUNA PLATFORM are not supported by the current version of FaceStream.

FaceStream changes

- Processing of streams accessible via HTTPS protocol has been supported.
- The stability of work with RTSP cameras has been improved by upgrading to a new version of FFMpeg.
- Now the “stream_id” parameter containing the stream ID is sent to the LUNA PLATFORM event.
- A new parameter “request_timeout_ms” has been added to the FaceStream settings, in which you can configure the timeout for sending and receiving events from/to the LUNA PLATFORM in milliseconds.

Default value - 30 seconds (30,000 milliseconds).

LUNA Streams changes

- A new parameter “timeout” has been added to the “[healthcheck](#)” section of stream management settings, which enables you to set the timeout in milliseconds for reading the encoded packet.
- The <service_name>_DB parameter section of the LUNA Streams database has been extended with a new optional parameter “dsn”, which specifies a DSN string that can contain various settings to control the connection to the database, such as multiple hosts, authentication data, port and others (settings depend on the database type).

Due to the implementation of a new setting, the classic settings for connecting to the database (“db_host”, “db_port”, “db_name”, “db_user” and “db_password”) in all settings of all services have become optional.

If necessary, you can combine the DSN string and the classic settings, but the DSN string is a higher priority. You can partially fill in the DSN string (for example, “postgres01,postgres02/luna_handlers”), and then the missing parameters will be filled in from the values of the parameters “db_host”, “db_port”, “db_name”, “db_user” and “db_password”.

After updating to the new version of LUNA PLATFORM, the “dsn” parameter will not appear in the “Settings” tab in the Configurator. To use DSN, you must manually specify the appropriate parameter. Below is an example of specifying the “dsn” parameter:

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

- The size of the “streams-configs” image has been reduced.
- The description of the automatic stream restart (“autorestart” section) has been expanded in the stream management settings.

See the section “Streams automatic restart” in the administrator manual.

- The “LUNA_STREAMS_LOGS_CLEAR_INTERVAL” section has been added to the LUNA Streams settings in the Configurator, which enables you to control the logic of automatically deleting stream logs from the “log” table of the LUNA Streams database.

Automatic deletion of logs helps to clear the database of a large number of unnecessary logs. The most recent entry for each stream is not deleted.

The following parameters are available:

- “interval” - Sets the interval for deleting logs.
- “interval_type” - Sets the interval type (weeks, days, hours, minutes, seconds).
- “check_interval” - Sets the frequency of checking logs for deletion (seconds).
- “active” - Enables/disables automatic deletion of thread logs.

Automatic log deletion is disabled by default.

The default settings include automatic deletion of logs (“active” = true) with checking of log streams in the database every 180 seconds (“check_interval” = 180) and delete logs older than 7 days (“interval” = 7 and “interval_type” = days).

Example of checking logs for deletion every 5 minutes and deleting logs older than 4 weeks:

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

- Support for [ECS](#) logging format has been added.

To use the new format, you need to set the value “ecs” in the “format” setting of the “LUNA_STREAMS_LOGGER” section.

When using the “ecs” value, the following fields will be used:

- “http.response.status_code” - Contains the HTTP response status code (e.g., 200, 404, 500, etc.).
- “http.response.execution_time” - Contains information about the time taken to execute the request and receive the response.
- “http.request.method” - Contains the HTTP request method (GET, POST, PUT, etc.).
- “url.path” - Contains the path in the request’s URL.
- “error.code” - Contains the error code if the request results in an error.

Fixed errors

- Fixed the error of generating an invalid json file with debugging results (enabled using the “save_debug_info” parameter).
- Fixed leaks and reduced FaceStream memory consumption.

FaceStream v.5.1.32

Changes

- FSDK has been updated from version 5.13.0 to version 5.15.1.

The FaceDetV3 detector has been updated in FSDK v.5.14.0.

Fixed errors

- Fixed the error where the “is_best_shot” field in the debug json logs (the “save_debug_info” parameter in the FaceStream settings) could contain an incorrect value of “false” when the Primary track policy was enabled.
- Fixed the error where the track could not be sent if a certain combination of parameters from the “primary_track_policy” and “liveness” sections of the stream management settings were enabled.

FaceStream v.5.1.30

Changes

- The display of track status logs will now be enabled by using the “debug” tag, rather than the “bestshot” tag.

Example of a track status log: I0417 17:03:09.311389 74 bodysbestshotHandler.cpp :134] [debug] Track [336363]. Status update ACTIVE

- In the container of the LUNA Streams service, the Python version has been updated to 3.11. Support for older versions of Python has been discontinued.

FaceStream v.5.1.28

Changes

- Now FaceStream will send unique “track_id” to LUNA PLATFORM in UUID format, for example, “671719eb-bee8-4cd4-8efd-da05000040”.

Fixed errors

- Fixed an error where two requests were made to the LUNA PLATFORM when sending the source frame with saving to the resource [“/images”] was enabled - request to the resource “/handlers/{handler_id}/events” with the bestshots and the source frame and an extra request to the resource “/images” with the source frame.

Now only one request is made to the resource “/handlers/{handler_id}/events” containing the bestshots and the source frame.

In addition, an error was fixed in which the “jpeg_quality_level” parameter (frame compression ratio) was ignored when making a request to the resource “/handlers/{handler_id}/events”.

FaceStream v.5.1.27

Changes

- Now the “image_origin_policy” mechanism of the LUNA PLATFORM handler will be used to save the source frame if the resource “/images” is supposed to be used.

Previously, several internal requests were made to save the source frame.

The address to the “/images” resource is set in the “frame_store” parameter in the streams management settings (for example, “frame_store”: “http://127.0.0.1:5000/6/images”).

This feature does not apply to saving source images to the LUNA Image Store storage bucket.

Note that in order to send the source frame, the “send_source_frame” parameter must be enabled in the FaceStream settings, and the handler policy must have the saving of source images enabled (parameter “policies” > “storage_policy” > “image_origin_policy” > “store_image” = “1”).

- The ability to specify ROI or DROI coordinates as a percentage of the current frame size has been added.

To achieve this, a new field called “mode” has been added to the parameters “roi” and “droi” in the “create stream”, “get stream” and “get streams” requests of the LUNA Streams service.

This field enables you to set one of two modes:

- “abs” - parameters “x”, “y”, “width” and “height” are set in pixels (old variant);
- “percent” - parameters “x”, “y”, “width” and “height” are set as percentages of the current frame size (new variant).

If the “mode” field is not specified in the request body, then the value “abs” will be used. Thus, there is no need to recreate the old streams.

- A new tag “estimator” has been added to the list of tags for logging (the “tags” parameter in the FaceStream settings), designed to collect and display statistics of completed estimates.

When the tag is enabled in the FaceStream logs, information is returned on the size of the batches with which the estimators were called. In future releases, the list of returned information may be expanded.

Example of log content with the “estimator” tag enabled:

```
[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%)
```

In this case, the statistics show the following:

- a total of 1311 calls were made to the “HeadPoseEstimator” estimator, which took 191 milliseconds;
- all calls (1311) were processed individually, each in its own batch of size 1;
- batch sizes from 2 to 16 were not used.

If the estimators were not called, then no information will be output in the FaceStream logs.

- The ability to disable filtering by AGS and pitch, yaw, roll angles has been added.

To disable AGS filtering, you must set the value “0” for the “min_score” parameter in the stream management settings.

To disable angle filtering, you must set the values “180” for the parameters “detection_pitch_threshold”, “detection_roll_threshold” and “detection_yaw_threshold” in the stream management settings.

- The GPU memory consumption of the FaceStream application has been reduced.

- The default FaceStream monitoring settings are now similar to the default LUNA PLATFORM monitoring settings.

This means that if the default settings are used for monitoring the LUNA PLATFORM, then to enable FaceStream monitoring, it is enough to enable the “send_data” parameter in the FaceStream settings. In this case, the FaceStream monitoring data will be saved to the “luna_monitoring” bucket.

- The section of activating the LUNA PLATFORM license for LUNA Streams has been moved to a separate license activation manual “FS_without_LP/FS_License_Activation_Manual_Standalone_Eng.pdf/html”.

This manual also includes information on activating the LUNA PLATFORM license using Guardant keys (see the LUNA PLATFORM v.5.45.1 release notes for more information).

FaceStream v.5.1.23

Changes

- Updated FSDK to version 5.13.0.

In this version of FSDK, the body detector has been updated to version 5.

Note that when updating the FSDK version, you need to pay attention to changes in the “TRACK_ENGINE_CONFIG” settings in the Configurator service. In this release, the “TRACK_ENGINE_CONFIG” settings have not changed.

- Now, when changing the frame resolution during stream processing, processing will be completed with the error frame size is changed from <value> to <value>.

If an automatic restart is set for stream (see the [autorestart](#) section of the “[create stream](#)” query), it will be restarted. In this case, FaceStream will again accept the stream for processing with a new frame resolution.

If the user is viewing the preview, user needs to refresh the preview window.

- Changed the order of applying estimators. Now the AGS estimator is applied first and then the HeadPose.
- Removed filtering by Liveness Head&Shoulders.

In the LUNA Streams OpenAPI specification, the `use_shoulders_liveness_filter` parameter is declared as **Deprecated** and will be ignored when trying to create a new stream.

The first value in the `livenesses_weights` parameter, which is responsible for filtering by Liveness Head&Shoulders, will also be ignored. For example, if you set the value `livenesses_weights = [0.1, 0.25, 0.75]`, the value 0.1 will be considered as 0.0.

Fixed errors

- Fixed an error that could cause FaceStream to crash when closing the application.

FaceStream v.5.1.20

Changes

- Now, along with sending the source frame to LUNA PLATFORM, a header `X-Luna-Meta-rescale` is sent containing the scaling factor of the source image in the form of a floating-point number.

Previously, when setting the width of the source frame for scaling (the `size_source_frame` parameter), it was impossible to determine the coordinates of the bbox on the source image, since only the coordinates of the bbox of the scaled image were sent to the LUNA PLATFORM.

To determine the real coordinates of the bbox on the scaled source image, it is necessary to divide the size of the obtained image by the obtained scaling factor.

Note. The real size is calculated with an absolute error of less than $1 / (2 * \text{scaling factor})$.

For example, if the obtained image has a size of 200x113, and the scaling factor is 0.104399, then the real size of the source image is 1916x1082 ($200/0.104399=1916$, $113/0.104399=1082$).

The scaling factor can be found in LUNA PLATFORM:

- in the response header to the “get image” request to the LUNA API service. The request must be made with the `with_meta = 1` query parameter;
- in the “image_id.meta.json” file located next to the source image in the LUNA Image Store bucket of source images.

The `X-Luna-Meta-rescale` header is only sent if sending the source frame is enabled (parameter `send_source_frame = 1`). The LUNA PLATFORM handler should also have a policy of saving the source image.

If scaling is not set for the source frame (parameter `size_source_frame = 0`), then the value of the `X-Luna-Meta-rescale` response header will be equal to 1.

- Section `<LUNA_STREAMS_HTTP_SETTINGS` was added to the LUNA Streams configurations, containing settings responsible for processing HTTP connections.

The following settings are available:

- `request_timeout` - the duration of time between the instant when a new open TCP connection is passed to the server, and the instant when the whole HTTP request is received. Value (in seconds) is integer number, default 60.
- `response_timeout` - the duration of time between the instant the server passes the HTTP request to the app, and the instant a HTTP response is sent to the client. Value (in seconds) is integer number, default 600.

- `request_max_size` - how big a request may be (bytes). Value (in bytes) is integer number, default 1gb.
- `keep_alive_timeout` - http keep alive timeout. Value (in seconds) is integer number, default 15.

See the following link for details: <https://sanic.dev/en/guide/deployment/configuration.html#builtin-values>

- New `FORMAT` setting was added to all LIM services, which enables you to specify the logging format - `default` (standard log output format) and `json` (json log output).

Fixed errors

- The error was fixed where enabling the `send_source_frame` parameter could lead to a decrease in FaceStream performance.
- The error was fixed in the Docker Compose script, due to which the uploading of LUNA Streams settings was performed through the script `load_dump.py` (the old method used in versions 5.1.12 and below), and not through the migration mechanism (the new method used in versions 5.1.13 and above).

Note. If the previous version of FaceStream was launched via Docker Compose, then when updating, you must additionally execute the command to upload settings via the migration mechanism for the previous version of LUNA Streams v.0.5.17, and then execute the same command to upload settings for the new version of LUNA Streams v.0.6.2. See the section “Migrate settings” FaceStream update manual.

FaceStream v.5.1.18

Changes

- Implemented functionality for collecting and sending FaceStream monitoring data to the Influx database. Monitoring for FaceStream is disabled by default. To enable FaceStream monitoring, you need to enable the “`send_data`” setting and set additional settings from the “monitoring” section of the FaceStream settings.

The following data is sent to InfluxDB:

- measurement element. It is equal to the value of `fs-requests`.
- tag set:
 - * `fs_ip` - IP address where FaceStream is deployed
 - * `source` - the “name” field set when creating a stream in LUNA Streams (optional)
 - * `stream_id` - stream ID
- field set:

- * track_id - track ID
- * event_id - event ID received from LUNA PLATFORM
- * request_id - external ID for communication with monitoring of LUNA PLATFORM services
- * track_start_time - track start time
- * track_best_shot_time - the time when the frame with the bestshot being sent appeared in the system
- * track_best_shot_min_size_time (optional) - the time when the detection size reached the value specified in the “best_shot_min_size” parameter
- * track_best_shot_proper_size_time (optional) - time when the detection size reached the value specified in the “best_shot_proper_size” parameter
- * liveness_start_time (optional) - Liveness start time
- * liveness_end_time (optional) - Liveness end time
- * bestshot_count - the number of bestshots sent
- * time_from_first_frame_to_send - the time that passed from the appearance of the first frame in FS to sending to LP
- * track_send_count - the number of tracks sent

Tags containing time are sent as UTC with microsecond precision.

- <timestamp> element. Is the time the bestshot(s) was(were) sent in microseconds.

See the “[Monitoring](#)” section of the administrator manual for more information.

- Added user interface for the LUNA Streams service.

The interface can be opened in a browser by specifying the address and port of the LUNA Streams service: <streams_server_address>:<streams_server_port>

The user interface of the service contains three tabs - “Streams”, “Groups” and “Queue”.

- “Streams” is a tab where streams statuses and their preview are displayed, it is possible to create and manage streams.
- “Groups” is a tab where streams groups are displayed.
- “Queue” is a tab where the streams that are in the processing queue are displayed.

See the detailed information in the “[LUNA Streams user interface](#)” section of the administrator manual.

- Implemented sending coordinates of bounding rectangles of faces (X, Y, width, height) when using an image without transformation (setting portrait_type=gost). Sending is implemented by adding the face_bounding_boxes parameter containing an array of bounding boxes corresponding to the detected faces.

The coordinates of the rectangle are defined in the portrait coordinate system.

The coordinates of the rectangle are saved in the `face_bounding_boxes` field of the event generation request body. These coordinates are used when generating an event, which allows face redetection, which is performed faster than face detection.

- The maximum size of batch with sets of images and detections sent for AGS and HeadPose estimation is limited to 16.

This reduces the memory consumption of the FaceStream.

Fixed errors

- Fixed the error where the value of `silent_period=-1` could cause the track to be sent several times.

This error caused multiple events with the same `track_id` to be generated in LUNA PLATFORM.

FaceStream v.5.1.16

Changes

- Implemented sending coordinates of bounding rectangles of faces (X, Y, width, height) when using a sample as an image source (setting `portrait_type=warp`). Sending is implemented by adding the `face_bounding_boxes` parameter containing an array of bounding boxes corresponding to the detected faces.

The coordinates of the bounding box are defined in the coordinate system of the source image.

Support for saving coordinates to an event is implemented in LUNA PLATFORM v.5.36.5. Thus, starting from this version of the FaceStream build, it is required to use a LUNA PLATFORM version of at least v.5.36.5.

The coordinates of the bounding box are stored in the `“face_detections” > “detection” > “rect”` field of the LUNA PLATFORM event.

- Manuals for installing and updating FaceStream were divided into two parts:
 - manuals for installing and updating FaceStream with LUNA PLATFORM launched. It is assumed that LUNA PLATFORM was launched according to the documentation from LUNA PLATFORM distribution where default paths are used. These manuals are located in the `“/docs”` directory.
 - manuals for installing and updating FaceStream without LUNA PLATFORM launched (with license activation and additional launch of Licenses, Configurator services and PostgreSQL and Influx databases). These manuals are located in the `“/docs/FS_without_LP”` directory.

FaceStream fixed errors

- The error was fixed where the `“callback-mode”` parameter of the TrackEngine settings was not read.

FaceStream only works with “callback-mode=0”. Attempts to set the value of “callback-mode=1” will result in a warning message in the FaceStream logs and the value of “callback-mode=0” will be used.

- Reduced generation time of data in JSON format and implemented sequential processing of frames.

This resulted in a fix for a error where using the “json” value to set “sending > request_type” would spike memory usage.

Note. Long time to generate JSON data can lead to increased memory consumption.

- The error was fixed in which the mode of sending images of detected faces without transformation (the “portrait_type=gost” parameter) didn’t work on the GPU.

LUNA Streams fixed errors

- The error was fixed where “location” > “geo_position” > “latitude”/“longitude” fields could not be set to integers.

Now you can specify both integer and float numbers.

FaceStream v.5.1.14

Changes

- FSDK was updated to version 5.12.0.

In this version, the Liveness Flying Faces estimator has been updated.

- The description of many parameters in the [LUNA Streams specification](#) has been expanded.

Fixed errors

- The error has been fixed where FaceStream could send an incorrect preview URL in the “preview > live > url” field of the LUNA Streams stream, which caused the following error and FaceStream continued to work:

```
Failed to validate input json' to occur in the FaceStream logs. Path: '
streams.0.preview.live.url', message: 'URL host invalid',
description: Bad/incomplete input data.
```

This error could occur due to the fact that LUNA Streams was started after FaceStream (for example, the server was restarted, where both FaceStream and LUNA Streams are running), and in order to determine its IP address, the FaceStream application needs to either connect to the LUNA Streams service, or read the predefined IP address in the system environment variable “VL_FACE_HOST” (see section “[Nuances of working with stream preview](#)” of the administrator manual).

Now if the IP address of FaceStream is determined by connecting to LUNA Streams and the service starts after FaceStream, then FaceStream will terminate with the error “Failed to get local IP address. Reason ...”.

If the IP address of FaceStream is determined using the correctly set “VL_FACE_HOST” environment variable, then such an error will not occur.

- The error has been fixed where the DROI bounding box flashed while viewing a stream in the browser if “[show_bbox](#)” was set to “0”.
- The error has been fixed where some detections with human body coordinates for the best shots were not sent to LUNA PLATFORM.

Accordingly, these detections were absent in events generated using the dynamic handler specified in the “detection_handler” parameter.

FaceStream v.5.1.13

Changes

- A mechanism for migrating LUNA Streams settings set in the LUNA Configurator service has been added.

LUNA Streams update

Previously, LUNA Streams settings were automatically migrated when migrating the LUNA Configurator service. Now LUNA Streams settings are migrated using a separate migration mechanism. See the upgrade manual for more details.

Launch LUNA Streams from scratch

Now the LUNA Streams settings are loaded into the Configurator service using the migrations mechanism. The FaceStream settings are still loaded into the LUNA Configurator using the “load_dump.py” script. See installation manual for more details.

Fixed errors

- FaceStream crash has been fixed that occurred when an incorrect address was specified in the “lunastreams” > “origin” parameter of the FaceStream settings.

Now the FaceStream operation is finished correctly, and the corresponding error is written to the logs.

- The error has been fixed where the valid range (from 0 to 1024 pixels) was not checked for the “size_source_frame” parameter of the FaceStream settings, resulting in the possibility of setting any values for this parameter.

- The error has been fixed where FaceStream would output a stack trace instead of an error if the port it was running on was busy.

Now the error `Error occurred on initializing application. Description: bind: Address already in use` is displayed in the FaceStream logs.

- The error has been fixed where changing the “enable_gpu_processing” setting of the FaceStream settings was not applied properly and caused a crash when the configuration was reloaded.

FaceStream v.5.1.12

Changes

- FSDK was updated to version 5.11.0.

Fixed errors

- The error was fixed that caused the stream error counter to be processed incorrectly. If during the “period” the number of errors did not reach the value “max_error_count”, then the error counter was not reset, but continued to increase.
- The error was fixed, due to which zero-length strings and “null” values from the “location” section of streams management settings were written to the LUNA PLATFORM event.

FaceStream v.5.1.10

Changes

- Support for the new LUNA PLATFORM 5 authorization system was added.

New optional authorization field “event_handler > authorization” was supported in the [stream creation settings](#), in which you need to specify “token” or “account_id”. The “event_handler” > “authorization” > “account_id” parameter must match the “account_id” parameter set when creating the stream. If the new authorization field is not filled in, the “account_id” set when creating the stream will be used.

When upgrading to new version of LUNA Streams, the database will be automatically migrated to support the new field.

Starting from this version of the FaceStream build, it is required to use the LUNA PLATFORM version at least v.5.32.0.

- LUNA Streams and FaceStream service containers are now named the same when manually launched and when launched with Docker Compose.

- The Python version was updated to 3.10 in the LUNA Streams and LUNA Configurator containers.
All commands related to using Python inside containers were updated in the documentation, namely, commands like “python3.9” were replaced with “python3”.
- New documents “FS_Upgrade_Manual_Eng.pdf” and “FS_Upgrade_Manual_Eng.html” were added to the distribution package, describing the upgrade process from the previous build of LUNA Streams and FaceStream services, as well as LUNA PLATFORM services launched in accordance with the FaceStream documentation.

Fixed errors

- The error was fixed where the “jpeg_quality_level” setting was not working correctly on the GPU.
With the “request_type” = “jpeg” value, only the first source frame was compressed. The rest of the source frames were not compressed.
With the “request_type” = “json” value, the source frames were not compressed, but all warps were compressed, except for the first one.

FaceStream v.5.1.9

FaceStream changes

- DROI calculation mechanism was changed.
Previously, DROI represented coordinates relative to ROI, taking into account image rotation. DROI now represents coordinates relative to the original frame.
- The FaceStream stream processing report regularly sent to LUNA Streams was expanded with new “last_frame” and “live” fields containing the URLs of the image’s last frame and the live stream.
These URLs can be received using specific requests to the LUNA Streams service (see below).
- FaceStream operation was accelerated and GPU memory consumption was reduced by limiting the estimation of five landmarks in cases where such estimation is not required.
- Displaying the IP address of the server running FaceStream was added to the FaceStream logs.

LUNA Streams changes

- Migration was added to LUNA Streams, in which all existing streams will be updated in accordance with the new DROI calculation mechanism (see FaceStream changes above).
When updating FaceStream, it is necessary to update LUNA Streams.
- A “include_luna_services” parameter was added to “[get health](#)” request.
The “include_luna_services” parameter allows you to enable or disable healthcheck for the LUNA PLATFORM services that the LUNA Streams service depends on. If this option is enabled, then additional requests are sent to the “/healthcheck” resources of these services.

- Two new resources were added - “[get last frame preview](#)” and “[get live preview](#)” that allow you to get the values of the “last_frame” and “live” fields from the FaceStream stream processing report (see FaceStream changes above).

New parameters “preview” > “last_frame” and “preview” > “live” were added to the “[get streams logs](#)” request.

- The automatic restart of the stream now has its own statuses (the “status” parameter).

The automatic restart statuses are listed below:

- “disabled” - stream automatic restart is disabled by user (“restart” parameter is disabled)
- “enabled” - automatic restart is enabled but is not currently active because the stream is not in the “failure” status
- “in_progress” - automatic restart in progress
- “failed” - the allowed number of automatic restart attempts was exceeded and none of the attempts were successful
- “denied” - automatic restart allowed by user, but not possible due to fatal error received in the FaceStream report. At the moment, the critical error has the following content: “Failed to authorize in Luna Platform”. In future versions, the list of critical errors may be updated.

The status of the automatic restart can be obtained using the “[get stream](#)” request.

FaceStream v.5.1.8

FaceStream changes

- The script for uploading FaceStream and LUNA Streams dumps to the LUNA Configurator service can now be run on Python 2.x and Python 3.x versions. To run the script on the Python 2.x version, you must use the “python” command. To run the script on Python 3.x, you must use the “python3” command. Also, the script for loading dumps was added to Docker Compose. Now, to use the “start_facestream.sh” script, you only need to have running LUNA PLATFORM containers and an activated license.
- Support for Liveness on the GPU was added.

FaceStream fixed errors

- The error was fixed where the “Streams total” counter in the FaceStream logs counted streams with a “failure” status.
- The error was fixed that was causing FaceStream crashes in a multi-threaded environment.
- The error was fixed where image rotation, controlled by the “rotation” setting, was performed clockwise on the CPU and counterclockwise on the GPU. Rotation is now clockwise on both types of processors.

- The error was fixed, in which FaceStream took more streams into processing than specified in the license key.
- The FaceStream crash was fixed that occurred when setting a non-existent tag in the “logging” > “tags” field in the FaceStream settings.
- The FaceStream crash was fixed that occurred when running on a GPU with “size_source_frame” from FaceStream settings set to non-zero.

LUNA Streams changes

- Significantly accelerated performing of requests to “[get stream](#)” and “[get streams](#)” resources with a large number of entries in the stream log tables of the LUNA Streams database.
- New resource “[delete streams logs](#)” was added, which enables you to delete the streams logs older than the date specified in the request (the “log_time__lt” parameter), excluding the last entry in the log.

FaceStream v.5.1.7

FaceStream changes

- Now the URL for sending the source frames of faces and bodies is specified in the “frame_store” parameter. Previously, the URL from the “origin” parameter was used to send the source frames of the bodies, and the URL to send the source frames of the faces was set in the “frame_store” parameter.
- Now, for failed requests to LUNA PLATFORM services, the FaceStream logs will show the “Luna-Request-Id” value generated by FaceStream.

Based on the generated “Luna-Request-Id” value, you can find an error in the logs of LUNA PLATFORM 5 services.

- A script “load_dump.py” was added to the “example-docker/luna_configurator/dumps/” directory of the distribution package, which enables you to upload FaceStream and LUNA Streams settings to LUNA Configurator.

Example of command to upload settings for LUNA Streams service: `python -m load_dump --dump-file=streams_dump.json --luna-config=http://127.0.0.1:5070/1`

This method is taken as the main one in the FaceStream installation manual (see the “Uploading settings to LUNA Configurator” section).

FaceStream fixed errors

- The crash was fixed that occurred due to incorrect handling of server errors when sending the source frame.

- The error was fixed, which led to a significant increase in RAM consumption during FaceStream operation.

LUNA Streams changes

- New parameters “duration” and “progress” were added to the “video_info” field of the response to a request to get information about stream (“[get stream](#)” request).

The “duration” parameter displays the total duration of the videofile in seconds, and the “progress” parameter displays the current videofile progress from 0 to 1, where 0 is the beginning of the videofile, 1 is the end of the videofile.

FaceStream v.5.1.6

FaceStream changes

- The “send_detection_path” parameter was removed from the FaceStream settings. Now the ability to send detections with the coordinates of the human body is controlled by the numerical value of the “minimal_body_track_length_to_send” parameter. If the “minimal_body_track_length_to_send” value is “0”, then no detections will be sent.
- Now, when sending a source frame to the LUNA Image Store using the “frame_store” parameter, the account ID will also be sent. The <source_frame_id>.meta.json file containing the “account_id” parameter will be saved in the folder with the source frame in the bucket.

LUNA Streams changes

- The error was fixed, in which after transferring the stream to the “pause” status, the status “in-progress” was displayed in the stream logs (see the “[get streams logs](#)” request).
- The error was fixed that allowed setting very large values for parameters in “[create stream](#)” and “[Post feedback on streams](#)” requests.

FaceStream v.5.1.5

FaceStream changes

- The ability to get the detection time (in seconds) of a face or body relative to the start of the video was added.

This time is stored in the “detect_ts” field of the LUNA PLATFORM event.

Getting face or body detection time is available only for the “videofile” source type.

- LUNA Streams version was updated.

FaceStream fixed errors

- The error was fixed where the stream would not go into “failure” status when invalid parameters were used.
- Getting the “detections_count” metric on the GET [“/metrics”](#) request was fixed.
- The “show_window” parameter, deprecated since version 5.0.6, was removed from the FaceStream settings.
- The error was fixed that occurs on some graphics cards when previewing a stream using the [“/streams/preview/{id}”](#) resource while using FaceStream with GPU.

LUNA Streams changes

- New “endless” parameter was added to the [“create stream”](#) and [“put stream”](#) requests of the LUNA Streams service, which enables you to control the restart of the stream when a network error is received (the error is determined by the system as an eof (end-of-file) marker).

The parameter is available only for the “udp” and “tcp” source types.

If the `endless` parameter takes the value `true`, then in case of receiving eof and successful reconnection, the processing of the stream will continue. If all reconnection attempts failed (see the [“healthcheck”](#) section), then the stream will take the “failure” status. If the parameter takes the value `false`, then the processing of the stream will not continue and the status of the stream will take the “done” status.

When using a video file as a “tcp” or “udp” source, it is assumed to use the value `false`. This will avoid re-processing an already processed fragment of the video file when receiving eof. If, when using a video file, the value of the `endless` parameter is `true`, then after the processing is completed, the video file will be processed from the beginning.

LUNA Streams fixed errors

- The order of getting stream logs in the [“get streams logs”](#) request was fixed. Now the new logs are shown first.

FaceStream v.5.1.4

Changes

- Two new launching keys for FaceStream were added - “CONFIG_RELOAD” and “PULLING_TIME”.

The “CONFIG_RELOAD” launching key enables checking for changes in the “FACE_STREAM_CONFIG” section of the LUNA Configurator service and takes the following values:

- “1” - change tracking is enabled, if there are changes in the configuration, all FaceStream containers will be automatically restarted;
- “0” - change tracking is disabled (default value).

The “PULLING_TIME” launching key sets the period for receiving new parameters from the “FACE_STREAM_CONFIG” section of the LUNA Configurator service in the range [1...3600] sec. Used in conjunction with the “CONFIG-RELOAD” tag. The default value is “10”.

Launching keys are also available in FaceStream mode with configuration files.

- New parameter “min_body_size_threshold” was added to the “filtering” section in the streams management settings.

The parameter sets the body detection size, less than which it will not be sent for processing. It is calculated as the square root of the product of the body detection height (in pixels) by its width (in pixels).

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

If the value is “0”, then filtering of body detection by size will not be performed.

Fixed errors

- The address of the source image is now recorded in the “detections” > “image_origin” field of the LUNA PLATFORM event. Previously, this address was recorded in the “face” > “user_data” field.
- The error was fixed, in which the “limit” parameter (maximum number of the streams to return) was not processed in the request to get streams from the queue (“/1/streams/processing/queue” resource) when filtering by group.

FaceStream v.5.1.3

Changes

- SDK was updated to version 5.6.0.
- The ability to group streams was added. Grouping is intended to combine streams with multiple cameras into logical groups. For example, you can group streams by territorial characteristic.

Any stream can belong to several groups, as well as none.

The group is created using the “[create group](#)” request. To create a group, you need to specify the required parameters “account_id” and “group_name”. If necessary, you can specify a description of the group.

Stream can be linked to a group in two ways:

- using the “group_name” or “group_id” parameters during stream creation (“[create stream](#)” request).
- using the “[linker](#)” request. In the request, you should specify the streams IDs and the group to which they need to be linked. You can also unlink streams from a group using this request.

If the stream was linked to a group, then the “[get stream](#)” or “[get streams](#)” requests will show the group in the “groups” field.

- The ability to transfer local track timestamps to events created by LUNA PLATFORM 5 was added.
 - For an event created by a static handler, the “create_time” and “end_time” parameters are now recorded.
 - For an event created by a dynamic handler, the “end_time” parameter is now recorded.
 - For the bestshots (samples), the “detect_time” parameter is now recorded.
- Now, when the “send_detection_path” parameter is enabled, the corresponding detection with body coordinates will be sent along with the bestshot by default. Previously, such a detection could not be sent.

FaceStream v.5.1.2

Changes

- The parameter “preferred_program_stream_frame_width” parameter was added to the streams management settings, intended to work with protocols that imply the presence of several channels with different bitrates and resolutions (for example, HLS).

If the stream has several such channels, then this parameter will enable you to select from all the channels of the whole stream the channel whose frame width is closer to the value specified in this parameter.

For example, there are 4 channels whose frame widths are 100, 500, 1000 and 1400. If the parameter “preferred_program_stream_frame_width” is equal to “800”, then a channel with a frame width of 1000 will be selected.

If the stream has only one channel, this parameter will be ignored.

This parameter is used only for “tcp” or “udp” types.

- The following new parameters was added to the “lunastreams” section of FaceStream settings:
 - request_stream_period - parameter sets the time period between requests to receive new streams from LUNA Streams in the range from 0.1 to 3600 seconds.
 - send_feedback_period - parameter sets the time period between sending reports on processed streams to LUNA Streams in the range from 1.0 to 3600 seconds. The value of this parameter should not exceed the value of the “STREAM_STATUS_OBSOLETING_PERIOD” parameter, set in the LUNA Streams service settings.
 - max_feedback_delay - parameter sets the maximum report sending delay in the range from 1.0 to 3600 seconds. If the report has not been sent within the given time, then FaceStream

will stop processing the current streams. The value of this parameter should not be less than the value of the parameter “send_feedback_period” and should not exceed the value of the parameter “STREAM_STATUS_OBSOLETING_PERIOD”, set in the LUNA Streams service settings.

FaceStream v.5.1.1

Fixed errors

- The error in the docker-compose.yml file that caused the error “host network_mode is incompatible with port_bindings” when running a Docker Compose script was fixed.

FaceStream v.5.1.0

Changes

- The principle of creating streams was changed. Previously, settings for streams were specified in the “input.json” configuration file or in the “FS_CONFIG” section of the Configurator. Now the settings for streams are set in the body of the JSON format and sent using an HTTP request to the LUNA Streams service (see the “[create stream](#)” request in the “StreamsReferenceManual.html” document of the distribution).

After sending an HTTP request with the given parameters to the LUNA Streams service, the parameters are added to the LUNA Streams database under the unique identifier “stream_id”. Depending on the number of available streams in the license (this information comes from the LUNA Licenses service), the stream parameters enters the internal queue, where it is in the “pending” status until FaceStream takes them from the queue for further processing (see the detailed description of the service in the “Interaction of FaceStream with LUNA Streams” in the administrator manual).

Licensing changes:

The service can process a certain number of streams simultaneously. The number of such streams is set in the LUNA PLATFORM 5 license key parameter, which determines the streams number for LUNA Streams operation. Thus, starting from version 5.1.0, FaceStream is licensed using the LUNA PLATFORM 5 key.

To launch LUNA Streams and FaceStream, you need at least three additional LUNA PLATFORM components:

- LUNA Licenses service. This service is used to license the LUNA Streams service.
- LUNA Configurator service. LUNA Streams settings are stored in this service in the same way as “FACE_STREAM_CONFIG” and “TRACK_ENGINE_CONFIG” settings for FaceStream. If you

- do not need to use LUNA Configurator, you can launch LUNA Streams using configuration files, but these steps are not described in the installation manual.
- PostgreSQL container. In this container, a LUNA Streams database is created, in which the stream parameters specified earlier in the JSON body are stored (see the description of the database in the “LUNA Streams database description” section of the administrator manual). Optionally, you can use an Oracle database. Its use is not described in the documentation.

The InfluxDB OSS 2 container is also launched by default. Using the Influx database, LUNA PLATFORM services are monitored. If necessary, monitoring can be disabled.

If the LUNA PLATFORM components are not launched and the LUNA PLATFORM license is not activated, then the following steps should be performed before launching FaceStream:

- License should be activated.
- Required components should be launched.
- LUNA Streams and FaceStream settings should be uploaded to LUNA Configurator.
- LUNA Streams database should be created and initialized.

If the LUNA PLATFORM components are launched and the LUNA PLATFORM license is activated, then the following steps should be performed before launching:

- LUNA Streams and FaceStream settings should be uploaded to LUNA Configurator.
- You should make sure that the license contains a parameter that determines the number of streams to be processed by the LUNA Streams service and specify the address of the server launched the LUNA Licenses service in the LUNA Configurator service (if the LUNA Licenses service is not on the server launched LUNA Streams).
- LUNA Streams database should be created and initialized.

For a description of how to perform these steps, see the FaceStream installation manual.

- The ability to deploy FaceStream using a Docker Compose script was added. The script enables you to automatically launch and configure LUNA Streams and FaceStream. Before running the script, the following steps should be performed:
 - License should be activated.
 - Required components should be launched.
 - LUNA Streams and FaceStream settings should be uploaded to LUNA Configurator.
 - LUNA Streams database should be created and initialized.
- Description of errors returned by the LUNA Streams and LUNA Licenses services, as well as general errors that may occur when interacting with the LUNA PLATFORM, was to the documentation.
- The “lunastreams” section was added to the FaceStream settings (“FACE_STREAM_CONFIG” section of the LUNA Configurator service), enabling you to specify the address of the LUNA Streams service used (“origin” parameter), its API version (“api_version” parameter) and the upper bound on the number of FS streams (“max_number_streams”). If LUNA Streams service

is launched on another server, then you should specify the current address of the service before launching FaceStream. See the FaceStream administrator manual for details.

- The names and structure of streams management settings was reworked. All settings are now divided into logical blocks. The following settings was renamed:
 - input, luna_account_id > account_id
 - input, transport > data, type
 - input, url > data, reference
 - output, image_store_url > event_handler, frame_store

The method of passing data to generate an event was reworked. Previously, to generate a LUNA PLATFORM event, the “output” > “url” parameter was used, in which, when working with faces, it was necessary to specify the address to the “/events” resource in the format `http://<luna_address>:<port>/6/handlers/<handler_id>/events`, where <handler_id> is the LP handler identifier. When working with bodies, separate parameters “luna_human_handler_id” and “luna_dynamic_human_handler_id” were used, and the “url” field was filled in the format `http://<luna_address>:<port>/6/`. Now the handler ID is set in the “event_handler” > “bestshot_handler”/“detection_handler” > “handler_id” group, where “bestshot_handler” is a static handler for a face or body, “detection_handler” is a dynamic “handler_id” LUNA PLATFORM for working with bodies (ex. “luna_dynamic_human_handler_id”). The LUNA API service address and API version are now set separately. Below is an example of passing data to generate an event when static and dynamic handlers are set for the body.

```
"event_handler": {
  "origin": "http://127.0.0.1:5000",
  "api_version": 6,
  "bestshot_handler": {
    "handler_id": "ee4c42b6-23ae-410e-a2aa-a4220e64ba4b"
  },
  "detection_handler": {
    "handler_id": "426542d6-5509-4e5b-8a01-e2abd5c0a8c6"
  }
}
```

If it is necessary to process a face, then “handler_id” should be set with parameters for processing a face in the “bestshot_handler” group, and the “detection_handler” group is optional.

See the FaceStream administrator manual for details on stream management settings.

- Support for normal FaceStream launch mode was dropped. Now you can launch FaceStream only in server mode.
- Information on launching FaceStream in server mode with configuration files was moved to the administrator manual (see “Use FaceStream with configuration files” section).

- The default logging parameters was changed in the FaceStream settings, in the “Logging” section:
 - The “severity” parameter is now equal to “1”. This means that only system warnings will be displayed in the logs.
 - The “mode” parameter is now equal to “l2c”. This means that logs will be output only to the console.

FaceStream v.5.0.7

Changes

- SDK was updated to version 5.5.0.
- The ability to get the current version of FaceStream using the GET “/version” request was added (see the detailed information in the “FaceStreamApi.html” document).
- Fixed an error where when running FS in server mode with Configurator, the “save-debug-info” parameter from the trackengine configuration was not used.
- Fixed an error where the source frame without detections with human body coordinates (detection_path) was not sent with the “send_source_frame” parameter set.
- Fixed an error where sending “track_id” to faces did not work, which is why the value of “track_id” was not recorded in the LUNA PLATFORM event.
- Fixed an error when using the mouth estimator. The error could occur when using two streams, when the first one used this estimator, and the second one did not.

FaceStream v.5.0.6

Changes

- FaceStream 5 is now distributed in Docker containers only. CentOS 8 is used for running FaceStream inside the container.
- FaceStream distribution package was updated. Now it includes documentation and configuration files required for the FaceStream launch.
- The “LP_Docker_Installation_Manual_Eng” document was added. It describes FaceStream launching in Docker container. The information is moved from the administrator manual. The document is provided in PDF and HTML formats.
- SDK was updated to version 5.4.1.
- Support for working with bodies was added. FaceStream searches for bodies in the stream and tracks them until they leave the frame or are overlapped.

There are additional settings for using FaceStream with bodies. Starting with FaceStream 5.0.6, all settings are divided into common and individual settings for faces and bodies in the documentation.

To enable body tracking, you need to change the “use-face-detector” parameter of the “trackengine.conf” configuration file to “use-body-detector” and also set individual settings for stream sources for bodies (see the relevant chapters).

The logic of creating events (sending bestshots and processing them by LUNA PLATFORM) for the body differs from the logic of creating an event for faces (see the administrator manual for details).

- All parameter names in configuration files were unified. For example, the names “frame-processing-mode” and “real_time_mode_fps” were used. Now all the names look the same: “frame_processing_mode” and “real_time_mode_fps”.
- Error messages were updated. Now they are more detailed and contain the path to the erroneous parameter.
- Synchronous waiting for a response from LUNA PLATFORM (also affects the performance of the working with bodies) was removed.
- The server time will now be sent with the bestshot. Previously, the device time was sent. This feature applies to all types of sources (video streams, video files and sets of images).
- Support for LUNA PLATFORM API versions 4 and 5 was discontinued. FaceStream cannot send data to LUNA PLATFORM 3, LUNA PLATFORM 4, Backport 3, and Backport 4 starting with this version. Sending images is only possible in LUNA PLATFORM 5 (API version 6).
- Using the “show_window” parameter and its associated parameters is no longer possible.