

---

# **FindFace**

***Release 1.6***

**NtechLab**

**Sep 07, 2023**



## SHOULD BE USEFUL:

<b>1</b>	<b>Glossary</b>	<b>3</b>
<b>2</b>	<b>Usage scenarios</b>	<b>5</b>
2.1	Vehicle scenarios . . . . .	5
2.1.1	License plate recognition only . . . . .	5
2.1.2	Recognition of multiple vehicles parameters . . . . .	5
2.2	Face scenarios . . . . .	6
2.2.1	Scenarios with PACS . . . . .	6
2.2.2	Time tracking . . . . .	6
2.2.3	Black and white lists scenarios . . . . .	6
<b>3</b>	<b>Features</b>	<b>7</b>
3.1	Event deduplication car face . . . . .	7
3.1.1	Possible scenarios of duplication . . . . .	7
3.1.2	How to configure event deduplication . . . . .	7
3.2	Spam events filtering car face . . . . .	8
3.2.1	Possible scenarios of spam events . . . . .	8
3.2.2	How to configure spam filtering . . . . .	8
3.3	Liveness face . . . . .	10
3.3.1	Possible scenarios of liveness detection . . . . .	10
3.3.2	How to configure liveness . . . . .	10
3.4	Headpose face . . . . .	11
3.4.1	Possible scenarios of headpose detection . . . . .	12
3.4.2	How to configure headpose . . . . .	12
<b>4</b>	<b>Introduction to Getting Started</b>	<b>13</b>
<b>5</b>	<b>STEP 1. CCTV Camera Requirements: characteristics and installation</b>	<b>15</b>
5.1	Face recognition . . . . .	15
5.1.1	CCTV Camera characteristics . . . . .	15
5.1.2	CCTV Camera installation . . . . .	16
5.2	Vehicle recognition . . . . .	17
5.2.1	CCTV Camera characteristics . . . . .	17
5.2.1.1	General characteristics . . . . .	17
5.2.1.2	Additional characteristics . . . . .	19
5.2.2	CCTV camera installation . . . . .	20
5.2.2.1	General installation recommendations . . . . .	20
5.2.2.2	Installation on barriers . . . . .	21
<b>6</b>	<b>STEP 2. Server and Software Requirements</b>	<b>23</b>
6.1	Server Requirements . . . . .	23

6.1.1	Image processing . . . . .	23
6.1.2	HD (720p) live streams processing . . . . .	23
6.1.3	FHD (1080p) live streams processing . . . . .	23
6.2	Software Requirements . . . . .	24
<b>7</b>	<b>STEP 3. Server preparation</b>	<b>25</b>
7.1	CPU server preparation . . . . .	25
7.1.1	Ubuntu OS . . . . .	25
7.1.2	CentOS . . . . .	26
7.2	GPU server preparation . . . . .	28
7.2.1	Ubuntu OS . . . . .	28
7.2.2	CentOS . . . . .	30
<b>8</b>	<b>STEP 4. FindFace Lite installer and the license uploading to the server</b>	<b>35</b>
8.1	Use scp command . . . . .	35
8.1.1	General scp syntax . . . . .	36
8.1.2	Scp syntax for login and password SSH authentication . . . . .	36
8.1.3	Scp syntax for secret and public SSH keys authentication . . . . .	36
8.2	Use an SFTP file manager . . . . .	37
<b>9</b>	<b>STEP 5. FindFace Lite installation</b>	<b>39</b>
9.1	Installation . . . . .	39
9.2	When the installation is finished . . . . .	41
<b>10</b>	<b>Config Settings</b>	<b>43</b>
10.1	How to work with config settings . . . . .	43
10.2	App configuration . . . . .	43
10.2.1	Settings possible values . . . . .	43
10.2.2	Settings description . . . . .	45
10.3	EAPI configuration . . . . .	46
10.4	License plate EAPI configuration . . . . .	46
10.5	VM configuration . . . . .	46
10.6	DB configuration . . . . .	46
<b>11</b>	<b>API</b>	<b>47</b>
11.1	Preparation to API usage . . . . .	47
11.2	API usage overview . . . . .	48
11.3	FindFace Lite API functionality . . . . .	49
<b>12</b>	<b>Edge Devices</b>	<b>51</b>
12.1	What are edge devices . . . . .	51
12.2	Preparation to recognition process . . . . .	51
12.2.1	Authenticate a device . . . . .	51
12.2.2	Create a Card . . . . .	52
12.2.3	Create an Object . . . . .	52
12.3	Edge Devices integration . . . . .	53
12.3.1	Create a Camera . . . . .	53
12.3.2	Configure the Edge Device . . . . .	54
<b>13</b>	<b>Webhooks</b>	<b>59</b>
13.1	Authenticate a device . . . . .	59
13.2	Create a Webhook . . . . .	59
	<b>Index</b>	<b>63</b>

The documentation consists of blocks. Use them to onboard to the *FindFace Lite* and to find the information fast.

Below there is the overview of the blocks:

- **Should be useful** block contains additional docs which can help you in work with the FindFace Lite.
- **Scenarios and features** block contains information about the cases of FindFace Lite usage and features that can help to improve the service performance.
- **Getting started** block lead you through the steps of preparation and installation of FindFace Lite.
- **Settings** block contains information articles describing what settings you can make in FindFace Lite.
- **Integration documentation** block includes articles about all integration methods and their descriptions.



**GLOSSARY****FindFace Lite**

FindFace Lite is a light version of the FindFace Multi.

**FindFace Lite Installer**

FindFace Lite is a file, containing a number of configs which install the FindFace Lite service.

**Identity authentication terminal**

Identity authentication terminal is an access control device with, which supports face authentication.

**VideoWorker**

VideoWorker is an interface object for tracking, processing and matching faces on multiple video streams.

**Liveness detection**

Liveness detection is a technique where an algorithm securely detects whether the source of a biometric sample comes from a fake representation or is a live human being. The biometric sample is a facial photo taken by a user.

**Event**

Event is a representation of an object (face or car) occurrence in the Camera frame. With active Camera Event is automatically created from VideoWorker detection data.

**Card**

Card is a profile of a real person or a car. Card can be one of two types: *face* or *car*.

**Camera**

Camera is a representation of any video stream that can also be a file.

**Object**

Object is a representation of particular face or car. To create it, you have to add the image and link this set to which Card it is linked.

**Webhook**

Webhooks are user-defined HTTP callbacks, triggering by an event in a web app.

**PACS**

PACS or ACS are a particular type of access control system used as an electronic security counter-measure. PACS can be used to control employee and visitor access to a facility and within controlled interior areas.

**Deduplication**

Deduplication is a feature that is used to prevent recognising of one person or a car as several different events in the period of time

**Antispam events**

Antispam is a feature that is used for distinguishing of a real Event to process and so called “spam” Event fixed in the system accidentally.

**Edge device**

Edge device is a physical device (e.g., Identity authentication terminals), which can connect and send images to FindFace Lite to recognise objects.

## USAGE SCENARIOS

FindFace Lite easily integrates with the enterprise systems and sends the processed data from connected cameras and access terminals.

Business logic of the systems remains with no changes, but at the same time enriches with the necessary video analytics data.

FindFace Lite usage scenarios are divided in two main categories:

- Vehicle scenarios
- Face scenarios

---

**Tip:** All scenarios can be improved by integrated FindFace Lite features, which are described in the Features article.

---

### 2.1 Vehicle scenarios

FindFace Lite processes video stream, recognizes vehicle parameters and sends an event to Physical Access Control System (PACS) through a webhook in the JSON format. Based on the information from FindFace Lite PACS performs a further access control scenario.

FindFace Lite suppose two options for interaction with PACS:

#### 2.1.1 License plate recognition only

If PACS makes decisions to grant access only based on a license plate number, and doesn't use vehicle parameters.

FindFace Lite sends an event to PACS with a license plate number, vehicle photo with license plate number, date, time and camera ID (used by PACS).

#### 2.1.2 Recognition of multiple vehicles parameters

Complex scenario which uses multiple parameters, besides license plate number:

Parameters	Scenario
Vehicle <b>front</b> and <b>rear</b> parts	To avoid false alarms when the camera sees the license plate on the car rear after passing the gate.
	To avoid false alarms for vehicle reverse.
Vehicle <b>type</b>	To meet municipal requirements for special vehicle access (e.g. ambulance car or fire truck).
License <b>plate number visi-</b> <b>bility</b>	To avoid false alarm when the vehicle is too far, or staying aside.

## 2.2 Face scenarios

FindFace Lite rules out all scam and false alarm cases using modern features such as **Liveness** and **Headpose**.

Moreover FindFace Lite can control medical mask presence and can be easily integrated with biometric access terminals.

### 2.2.1 Scenarios with PACS

FindFace Lite processes videostream, identifies person parameters and sends an event to PACS through a webhook in the JSON format.

Based on the information from FindFace Lite PACS performs a further access control scenario.

### 2.2.2 Time tracking

FindFace Lite allows to collect working time data from the Chrome application on different devices and easily make an export to working hours accounting systems (ERP, WFM).

Workers can easily mark start and finish time with just pressing the button and looking at the camera. Advanced face falsification security system Liveness prevents all the false alarms.

### 2.2.3 Black and white lists scenarios

FindFace Lite can be integrated with CRM and security systems in order to perform access control scenarios:

- When a client or a person from a white list comes, FindFace Lite can interact with customer loyalty program, send to CRM (or any other system) an event with a person ID and all the other parameters from the person card.
- When person from a black list comes, FindFace Lite will send an event to external system which perform necessary action.
- FindFace Lite can control medical mask presence within the staff and register an event with all the staff connected data (image without mask, staff ID etc.)

## FEATURES

### 3.1 Event deduplication car face

**Deduplication of events** is a feature that is used to prevent recognising of one person or a car as several different events in the period of time and simplifies integration with PACS that do not support deduplication.

---

**Note:** Deduplication works for events from all connected cameras.

---

It means that in FindFace Lite you can set a period of time during which Events with the same person or a car will be considered as duplicates and the system will react and process only on the first one and the following duplicating events will be ignored.

#### 3.1.1 Possible scenarios of duplication

- Events with **Car** can be duplicated if the vehicle stops for a while in front of the barrier or if the rear license plate of a vehicle that has already passed through the barrier is recognized by the exit camera.
- Events with **Face** can be duplicated if the camera fixed the face of the person that has already passed through the PACS who showed face to the camera again for any reason.

More scenarios are described in *Findface Lite scenarios article*.

#### 3.1.2 How to configure event deduplication

1. Open the configuration file located in the **FFlite -> api\_config.yml** using one of the editors (e.g., nano command).
2. Enable the feature by setting the **dedup\_enabled** parameter to *true*.
3. Configure the savings settings for duplicates in the **save\_dedup\_event** parameter, choose *true* to save duplicates and *false* to not to save.
4. Set the parameters for **car** and **faces** recognition separately.

Parameter	Default value	Description
<b>face_dedup_confidence</b>	0.9	Confidence of matching between 2 car or face Events to be considered duplicates. If matching score is equal or more than set value, Event is labeled as duplicating.
<b>car_dedup_confidence</b>	0.9	
<b>face_dedup_interval</b>	1	Time interval in seconds during which Events with the same car or face will be considered as duplicates.
<b>car_dedup_interval</b>	1	

5. Save the changes in the file and close the editor.
6. Apply new settings by restarting the `api` service with the following command:

```
docker compose restart api
```

---

**Tip:** Read the detailed instruction about all configuration settings in the [article](#).

---

## 3.2 Spam events filtering car face

**Spam events filtering** is a feature that is used for distinguishing of a real Event to process and so called “spam” Event fixed in the system accidentally.

It means that in the moment of creating an Event from added Camera or sending an Event from the [edge device via POST request](#) you can set the image detection area and omit its spamming part.

### 3.2.1 Possible scenarios of spam events

- Events with **Car** can be spam events if a vehicle is parked near the barrier and the license plate fall into the field of view of the video camera multiple events will be generated.

If the license plate is found in the PACS database, then the barrier will be opened.

- Events with **Face** can be spam events if a person stay near the camera, but is not going to pass through the PACS. If the person is found in the PACS database, the barrier will be opened.

More scenarios are described in [Findface Lite scenarios article](#).

### 3.2.2 How to configure spam filtering

Settings of capturing frames is configured in the **roi** parameter, where you have to specify the distance from each frame or image side.

---

**Tip:** Currently settings are only available in via API requests

---

- To configure the **roi** setting for **videostreams**, use the following requests and the format **WxH+X+Y** for the **roi** parameter:
  - POST `/v1/cameras/` to create the Camera object.
  - PATCH `/v1/cameras/{camera_id}` to the update Camera object.

```

{
  "name": "test cam",
  "url": "rtmp://example.com/test_cam",
  "active": true,
  "single_pass": false,
  "stream_settings": {
    "rot": "",
    "play_speed": -1,
    "disable_drops": false,
    "ffmpeg_format": "",
    "ffmpeg_params": [],
    "video_transform": "",
    "use_stream_timestamp": false,
    "start_stream_timestamp": 0,
    "detectors": {
      "face": {
        "roi": "1740x915+76+88", <<-- roi
        "jpeg_quality": 95,
        "overall_only": false,
        "filter_max_size": 8192,
        "filter_min_size": 1,
        "fullframe_use_png": false,
        "filter_min_quality": 0.45,
        "fullframe_crop_rot": false,
        "track_send_history": false,
        "track_miss_interval": 1,
        "post_best_track_frame": true,
        "post_last_track_frame": false,
        "post_first_track_frame": false,
        "realtime_post_interval": 1,
        "track_overlap_threshold": 0.25,
        "track_interpolate_bboxes": true,
        "post_best_track_normalize": true,
        "track_max_duration_frames": 0,
        "realtime_post_every_interval": false,
        "realtime_post_first_immediately": false
      }
    }
  }
}

```

- To configure the **roi** setting for **images** got form an edge device, use the following request and the format [**left, top, right, bottom**] for the **roi** parameter:
  - POST /v1/events/{object\_type}/add to post the Event object.

```

{
  "object_type": "face",
  "token": "change_me",
  "camera": 2,
  "fullframe": "somehash.jpg",
  "rotate": true,
  "timestamp": "2000-10-31T01:30:00.000-05:00",

```

(continues on next page)

(continued from previous page)

```
"mf_selector": "biggest",  
"roi": 15,20,12,14 <-- roi  
}
```

### 3.3 Liveness face

**Liveness** is a technology used in CCTV cameras to determine whether the biometric trait presented to the system belongs to a live person or is a spoof attempt such as a photograph, a video, or a mask.

FindFace lite uses passive method of liveness detection, that uses algorithms to analyze various features of a person's face, such as pupil movement, facial expressions, or the presence of micro-movements to determine if the person is alive.

This method is less invasive than other liveness detection methods and:

- it does not require any action from the user. This can lead to higher user acceptance rates and fewer errors due to user discomfort or error;
- it is real-time: it can quickly and accurately authenticate a user without causing any delay or disruption to the authentication process.
- it does not require any additional hardware or sensors;
- it is difficult to spoof than other types of liveness detection, such as those that require the user to perform a specific action.

#### 3.3.1 Possible scenarios of liveness detection

Liveness detection can be used in lots of scenarios, there are several of them:

- **Banking** – Liveness detection may be used in banking to verify the identity of customers.

For example, a customer may be required to present their face to a camera during a video call with a bank representative, and the system can use liveness detection to ensure that the customer is alive and not presenting a fake photo.

- **Employee Access Control**: Liveness detection can be used to control employee access to secure areas in the workplace.

For example, if an employee attempts to enter a secure area by presenting a photograph or a mask, the CCTV camera with liveness detection can deny access and notify the security team.

#### 3.3.2 How to configure liveness

1. Open the configuration file located in the **FFlite** -> **api\_config.yml** using one of the editors (e.g., nano command).
2. Enable the feature by adding the **liveness** value to the **face\_features** parameter.
3. Configure the source liveness detection in the **liveness\_source** parameter.
  - If the detection source is going to be an image (Events will be created via POST `//object_type/add`), set the **eapi** value.
  - If the detection source is going to be a videostream (Events will be created in FindFace Lite automatically after recogniton from a videstream), set the **vw** value.

4. Save the changes in the file and close the editor.
5. Apply new settings by restarting the `api` service with the following command:

```
docker compose restart api
```

**Tip:** Read the detailed instruction about all configuration settings in the [article](#).

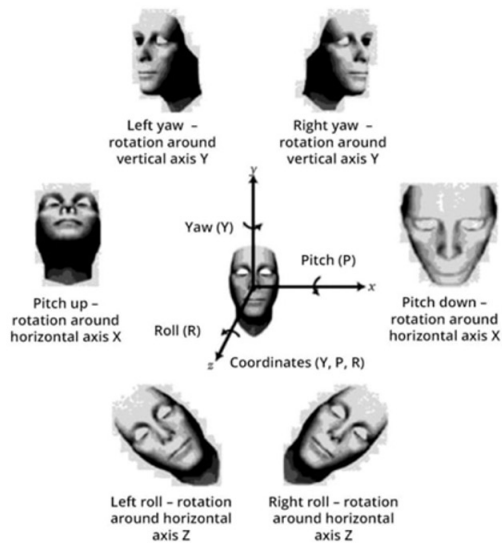
## 3.4 Headpose face

Headpose feature refers to the ability of the camera to detect and track the orientation and movement of a person's head relative the CCTV camera in real-time.

**Warning:** The headpose feature does not work if the person wear a medmask.

To detects the headpose the in two-dimensional space the FindFace Lite uses pitch and yaw.

- Pitch refers to the rotation of the head around its horizontal axis, which runs from ear to ear. Positive pitch indicates that the head is tilted forward, while negative pitch indicates that the head is tilted backward.
- Yaw refers to the rotation of the head around its vertical axis, which runs from top to bottom. Positive yaw indicates that the head is turned to the right, while negative yaw indicates that the head is turned to the left.



### 3.4.1 Possible scenarios of headpose detection

Headpose detection can be used in various scenarios where face recognition is used to improve accuracy and security, there are several of them:

- Improving employee access control systems by ensuring that the face of the employee matches the expected orientation.

It means that if a person stay near the camera, turns the head to the camera, but is not going to pass through the PACS, the access will not be provided, because of detected headpose.

- Improving the comfort, while using the PACS system. If an employee approaches a security checkpoint at an awkward angle, the CCTV camera with headpose detection can trigger an alert to the access control system to reposition the camera and ensure proper orientation of the face.

### 3.4.2 How to configure headpose

1. Open the configuration file located in the **FFlite** -> **api\_config.yml** using one of the editors (e.g., **nano** command).
2. Enable the feature by adding the **headpose** value to the **face\_features** parameter.
3. Save the changes in the file and close the editor.
4. Apply new settings by restarting the **api** service with the following command:

```
docker compose restart api
```

---

**Tip:** Read the detailed instruction about all configuration settings in the [article](#).

---

## INTRODUCTION TO GETTING STARTED

**Getting Started** block contains 5 steps, going through which you will easily install *FindFace Lite*.

**For the correct work of the service you have to:**

- Prepare a CCTV camera according recommendations (*STEP 1*).
- Prepare a server according necessary characteristics (*STEP 2*).
- Prepare it installing Docker Engine and Docker Compose (*STEP 3*).

**After all preparation steps are behind:**

- Upload the FindFace Lite installer and the service license to the server (*STEP 4*).
- Install FindFace Lite (*STEP 5*).

**Good luck!**

If you have questions, please, do not hesitate to write us on **support@ntechlab.com**.



## STEP 1. CCTV CAMERA REQUIREMENTS: CHARACTERISTICS AND INSTALLATION

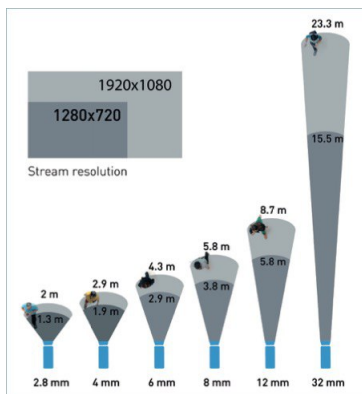
### 5.1 Face recognition

#### 5.1.1 CCTV Camera characteristics

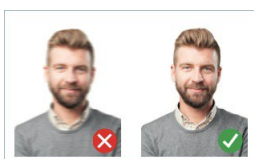
1. The minimum pixel density required for identification is 500 pixels/m (roughly corresponds to a face width of 80 pixels).



2. Select such a focal length of the camera's lenses that provides the required pixel density at a predetermined distance to the recognition objects. The picture below demonstrates how to calculate the focal length subject to the distance between the camera and recognition objects. Estimating the focal length for a particular camera requires either calculators or a methodology provided by the camera manufacturer.



3. The exposure must be adjusted so that the face images are sharp ("in focus"), non-blurred, and evenly lit (not overlit or too dark).



4. For imperfect lighting conditions such as flare, too bright or too dim illumination, choose cameras with WDR hardware (Wide Dynamic Range) or other technologies that provide compensation for backlight and low illumination. Consider BLC, HLC, DNR, high optical sensitivity, Smart infrared backlight, AGC, and such.



5. Video compression: most video formats and codecs that `FFmpeg` can decode.
6. Video stream delivery protocols: RTSP, HTTP.

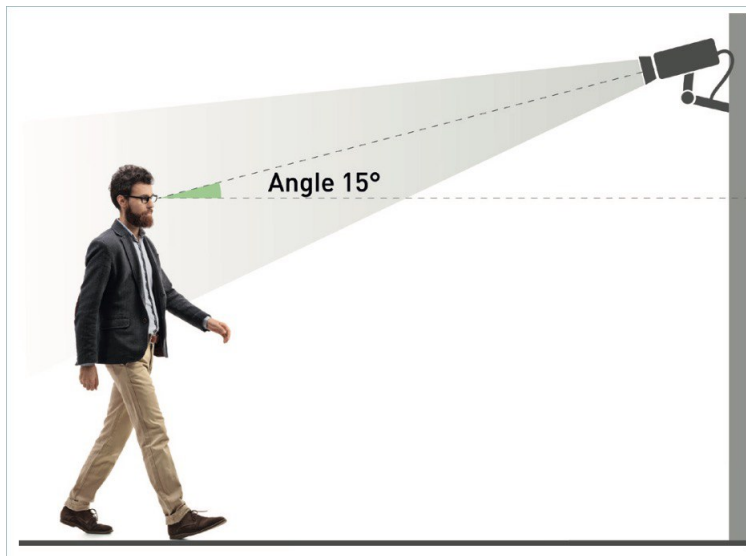
---

**Tip:** To calculate the precise hardware configuration tailored to your purposes, contact our experts by [support@ntechlab.com](mailto:support@ntechlab.com).

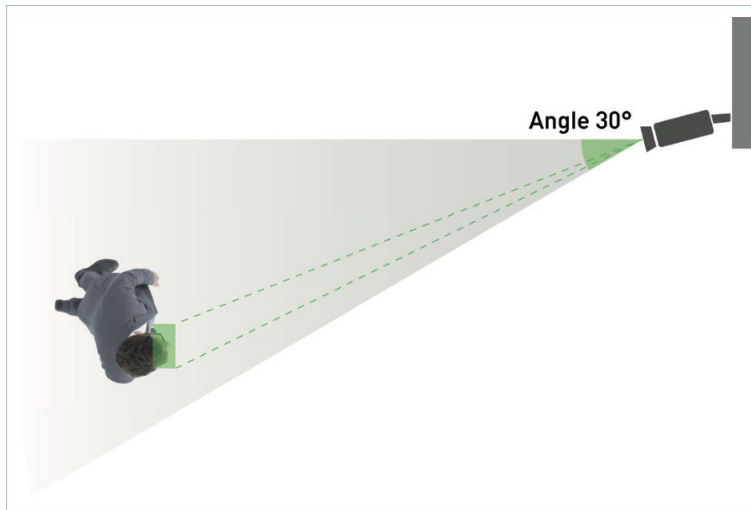
---

### 5.1.2 CCTV Camera installation

1. For correct face detection in a video stream, mount the camera so that the face of each individual entering the monitored area surely appears in the camera field of view.
2. The vertical tilt angle of the camera should not exceed  $15^\circ$ . The vertical tilt is a deviation of the camera's optical axis from the horizontal plane, positioned at the face center's level for an average height person (160 cm).



3. The horizontal deflection angle should not exceed  $30^\circ$ . The horizontal deflection is a deviation of the camera's optical axis from the motion vector of the main flow of objects subject to recognition.



## 5.2 Vehicle recognition

### 5.2.1 CCTV Camera characteristics

#### 5.2.1.1 General characteristics

FindFace lite requires the configuration described in the tables below.

#### Object in frame requirements

Parameter	Minimal requirements	Recommended requirements
Object size: vehicle width	$\geq 80$ px	$\geq 120$ px
Object size: license plate number width	$\geq 100$ px	$\geq 150$ px
Object size: LPN + vehicle	$\geq 340$ px	$\geq 340$ px
Object allowable overlap	$\leq 30\%$	$\leq 15\%$

## Camera requirements (for a digital image)

Parameter	Minimal requirements	Recommended requirements
Matrix size	$\geq 1/2,8$	$\geq 1/1,8$
Focal length	$\geq 1,5$ mm	$\geq 4$ mm
Light sensitivity (color)	$\leq 0.1$ lux	$\leq 0,05$ lux
TCP protocol	Yes	Yes
Broadcast resolution	$\geq 720 \times 576$	$\geq 1920 \times 1080$
Broadcast quality	3000-4000 kb/s	$\geq 4000$ kb/s
Frame rate	$\geq 15$	$\geq 50-60$
Shutter speed	up to 1/100	up to 1/500
H.264 support	H.264	H.264, H.265
Keyframe frequency adjustment	Yes	Yes
WDR support	Yes	Yes (up to 120 dB)
Aperture adjustment	Not required	Yes
Focal length adjustment	Not required	Yes
Mechanical IR filter	Not required	Yes
ONVIF support	Not required	Yes

## Camera mounting (object in a frame allowable rotation)

Parameter	Minimal requirements	Recommended requirements
Camera vertical angle	$\leq 45^\circ$	$\leq 30^\circ$
Camera horizontal tilt angle (vehicle)	Not matter	Not matter
Camera horizontal tilt angle (LPN)	$\leq 30^\circ$	$\leq 15^\circ$

## Lighting requirements

Parameter	Minimal requirements	Recommended requirements
Illumination in the recognition zone	$\geq 150$ lux	$\geq 200$ lux
Backlight compensation	$\leq 200$ lux	$\leq 100$ lux

## Broadcast camera settings requirements

Settings	Parameter	Recommended requirements
Exposition	Iris mode	auto
	Auto iris level	50
	Exposure time	1/200
	Gain	25
Camera focus	Camera focus	Should be configured manually for a specific scene
Backlight settings	BLC	OFF
	WDR	OFF
	HLC	ON (for barriers camera, if collecting of vehicles attributes is not required)

### 5.2.1.2 Additional characteristics

#### IR illumination

Highly recommended option for camera. License plate number effectively reflects lights and in low light conditions the light from camera IR will perfectly highlight the car number.

---

**Note:** IR illumination characteristics of the equipment written by the manufacturer (e.g. 10, 20, 50 meters) are the range to the complete backlight extinction.

So, the effective range is usually 30-40% lower. Please, keep that in mind.

---

#### SmartIR

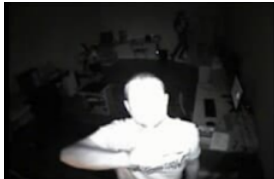
If there is an IR illuminator in the camera, SmartIR option could greatly improve the quality of image in low light conditions.

SmartIR or smart IR illumination control allows you to reduce the backlight intensity if the subject is too close and the frame is overexposed.

*With Smart IR*



*Without Smart IR*



#### WDR (Wide Dynamic Range)

Use the WDR setting to adjust the white balance.



### BLC (Blacklight Compensation)

Use the BLC setting to correct backlight problems.

### HLC (Highlight Compensation)

Use the HLC setting to compensate for overexposed areas. HLC automatically detects redundant light sources and reduces flare, greatly improving the clarity of bright areas.

When HLC is activated, the camera will proceed bright areas such as a spotlight and adjust the exposure accordingly.

With HLC the camera will try to build the whole scene exposure correctly, reducing the brightness of overlighted areas.



## 5.2.2 CCTV camera installation

Objects in the frame should be detailed, focused, not blurred and highly contrasted.

For proper analytics follow the mounting guidelines below.

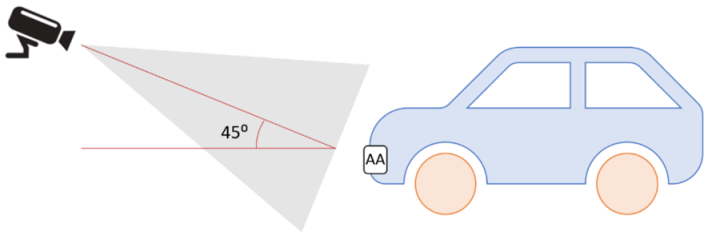
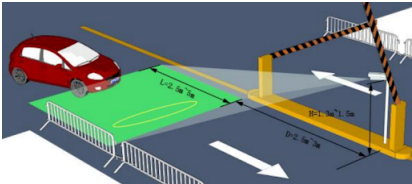
### 5.2.2.1 General installation recommendations

- **The distance from camera to recognition zone is arbitrary.** Cameras with appropriate lenses are being selected depending on the distance.
- **Camera should be mounted to a fixed rigid construction.**
- **Avoid sunlight or excess light** in camera lens, it can lead to image flare.
- **Object in the frame** should be completely visible. The central camera axis should approach to recognition zone center, so the object itself is in the center of the frame.
- **Resolution and camera lens** should be clear and not blurry, without any visible distortion.

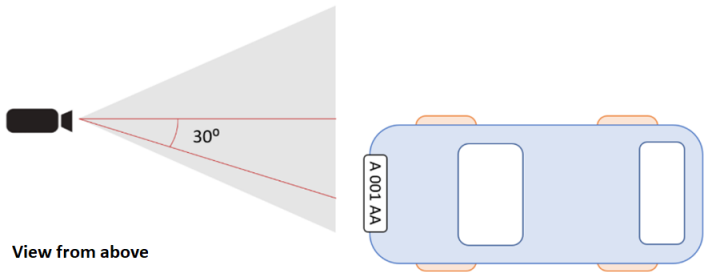


5.2.2.2 Installation on barriers

Parameters	Requirements
Installation height (H)	1.3-1.5 m
Distance from camera to image corner (D)	2.5 - 3 m.
Visible frame distance (L)	2.5-5m



Side view



View from above



## STEP 2. SERVER AND SOFTWARE REQUIREMENTS

### 6.1 Server Requirements

*FindFace Lite* requires different server configuration depending on the type of processed information (image or stream) and on quantity of used camera and *identity authentication terminals*.

#### 6.1.1 Image processing

For image processing, when *identity authentication terminal* detects a face and sends a result to FindFace Lite, server requirements are the following:

Devices Quantity	Physical Intel Cores >2.4 GHz	RAM, GB
1 – 8	4	6
8 – 16	4 – 6	8
16 – 24	4	8

#### 6.1.2 HD (720p) live streams processing

For detection while processing HD (720p) live streams (20 FPS) server requirements are the following:

Devices Quantity	Physical Intel Cores >2.4 GHz	RAM, GB
1	4	8
5	8	10
10	14	16

#### 6.1.3 FHD (1080p) live streams processing

For detection while processing FHD (1080p) live streams (20-25) FPS server requirements are the following:

Devices Quantity	Physical Intel Cores >2.4 GHz	RAM, GB
1	6	8
5	16	10
10	24	16

## 6.2 Software Requirements

Software	Recommendations
Operation System	Ubuntu 18.04 x64, CentOS 7 and other similar OS.
Command Line	Linux command line only.
Docker Engine	Version 19.03+
Docker Compose	Version 2.2.3+
To set up Docker software correctly, please, read <a href="#">Step 3</a> .	
NVIDIA Container Toolkit	Only for <b>GPU server</b> . Version 1.7.0+ (nvidia-docker2 >= 2.8.0)

## STEP 3. SERVER PREPARATION

### 7.1 CPU server preparation

To prepare a CPU server to FindFace Lite, please, install [Docker Engine \(19.03+\)](#) and [Docker Compose \(2.2.3+\)](#) .

Before you install [Docker Engine \(19.03+\)](#) and [Docker Compose \(2.2.3+\)](#) for the first time on a new host machine, you need to set up the Docker repository.

Steps from setting up the **Docker repository** to **Docker and Docker Compose installation** are described in the guides for Ubuntu OS and CentOS below.

#### 7.1.1 Ubuntu OS

1. Update apt and install packages for data encryption to use the repository over HTTPS:

```
sudo apt-get update
sudo apt-get install \
    ca-certificates \
    curl \
    gnupg \
    lsb-release
```

2. Add the GPG key given by Docker. Use the command below:

```
sudo mkdir -p /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -
  ↳o /etc/apt/keyrings/docker.gpg
```

3. Add the Docker repository using the following command:

```
echo \
"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.
  ↳gpg] https://download.docker.com/linux/ubuntu \
$(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /
  ↳dev/null
```

4. Update the apt:

```
sudo apt-get update
```

5. Install **Docker** and **Docker Compose Plugin**:

```
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-compose-  
→plugin
```

6. Check on how the **Docker** was installed using the command which downloads a test image and runs it in a container:

```
sudo docker run hello-world
```

The container will be started, you will see the operation success message, then the container will automatically be stopped.

```
> docker run hello-world  
Unable to find image 'hello-world:latest' locally  
latest: Pulling from library/hello-world  
2db29710123e: Pull complete  
Digest: sha256:faa03e786c97f07ef34423fccceec2398ec8a5759259f94d99078f264e9d7af  
Status: Downloaded newer image for hello-world:latest  
  
Hello from Docker!  
This message shows that your installation appears to be working correctly.  
  
To generate this message, Docker took the following steps:  
1. The Docker client contacted the Docker daemon.  
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.  
   (amd64)  
3. The Docker daemon created a new container from that image which runs the  
   executable that produces the output you are currently reading.  
4. The Docker daemon streamed that output to the Docker client, which sent it  
   to your terminal.  
  
To try something more ambitious, you can run an Ubuntu container with:  
$ docker run -it ubuntu bash  
  
Share images, automate workflows, and more with a free Docker ID:  
https://hub.docker.com/  
  
For more examples and ideas, visit:  
https://docs.docker.com/get-started/
```

7. Make sure the **Docker Compose Plugin** is also installed correctly. Use the following command to check:

```
docker compose version
```

```
> docker compose version  
Docker Compose version v2.2.3
```

Now all requirements for FindFace Lite are followed, please, go to the next [STEP](#) to upload FindFace Lite installer and license on the server.

## 7.1.2 CentOS

1. Install the yum-utils package and set up the repository. Use the command below:

```
sudo yum install -y yum-utils  
sudo yum-config-manager \  
  --add-repo \  
  https://download.docker.com/linux/centos/docker-ce.repo
```

2. Install **Docker** and **Docker Compose Plugin**:

```
sudo yum install docker-ce docker-ce-cli containerd.io docker-compose-plugin
```

**Note:** If prompted to accept the GPG key, verify that the fingerprint matches **060A 61C5 1B55 8A7F 742B 77AA C52F EB6B 621E 9F35** value, and if so, accept it.

3. Start Docker using the following command:

```
sudo systemctl start docker
```

4. Check on how the **Docker** was installed using the command which downloads a test image and runs it in a container:

```
sudo docker run hello-world
```

The container will be started, you will see the operation success message, then the container will automatically be stopped.

```
> docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
2db29710123e: Pull complete
Digest: sha256:faa03e786c97f07ef34423fccceec2398ec8a5759259f94d99078f264e9d7af
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/
```

5. Make sure the **Docker Compose Plugin** is also installed correctly. Use the following command to check:

```
docker compose version
```

```
> docker compose version
Docker Compose version v2.2.3
```

Now all requirements for FindFace Lite are followed, please, go to the next *STEP* to upload FindFace Lite installer and license on the server.

## 7.2 GPU server preparation

To prepare a GPU server to FindFace Lite, please, install [NVIDIA Container Toolkit](#).

Before you install NVIDIA Container Toolkit for the first time on a new host machine, you need to prepare server according requirements:

- NVIDIA Linux drivers  $\geq 418.81.07$  (note that older driver releases or branches are unsupported, to install drivers go to the official [NVIDIA guide](#));
- NVIDIA GPU with Architecture  $\geq$  Kepler;
- Docker  $\geq 19.03$ .

Steps from checking GPU server configuration to the **Docker software** and **NVIDIA Container Toolkit** installation are described in the guides for Ubuntu OS and CentOS below.

### 7.2.1 Ubuntu OS

1. Check the GPU drivers version using the command below:

```
nvidia-smi
```

**Driver Version:** should be  $\geq 418.81.07$ .

```
> nvidia-smi
Sun Dec 11 18:38:18 2022
```

NVIDIA-SMI 460.80 Driver Version: 460.80 CUDA Version: 11.2									
GPU	Name	Persistence-M	Bus-Id	Disp.A	Volatile	Uncorr.	ECC		
Fan	Temp	Perf	Pwr:Usage/Cap	Memory-Usage	GPU-Util	Compute M.	MIG M.		
0	GeForce RTX 208...	Off	00000000:01:00.0	On	24%	Default	N/A		
43%	45C	P8	27W / 300W	668MiB / 11016MiB			N/A		

Processes:							
GPU	GI ID	CI ID	PID	Type	Process name	GPU Memory Usage	
0	N/A	N/A	1966	G	/usr/lib/xorg/Xorg	24MiB	
0	N/A	N/A	2218	G	/usr/bin/gnome-shell	83MiB	
0	N/A	N/A	2605	G	/usr/lib/xorg/Xorg	403MiB	
0	N/A	N/A	13798	G	...474575672989714402,131072	64MiB	
0	N/A	N/A	15793	G	...AAAAAAAA= --shared-files	28MiB	
0	N/A	N/A	15836	G	alacritty	11MiB	
0	N/A	N/A	23349	G	alacritty	11MiB	
0	N/A	N/A	25405	G	...RendererForSitePerProcess	3MiB	

If it is not so, please, go to the official [NVIDIA guide](#) to install the drivers.

2. Check a graphics card model using command below:

```
nvidia-smi -L
```

3. Verify that graphics card has architecture  $\geq$  **Kepler**. Find your graphics cards model in the list below and check:

Architecture (from the oldest to the newest)	Series
<b>Fermi</b>	GeForce 400 and 500: GTX 480, GTX 470, GTX 580, GTX 570;
<b>Kepler</b>	GeForce 600 and 700: Nvidia GTX 680, 670, 660, GTX 780, GTX 770;
<b>Maxwell</b>	GeForce 900: GTX 960, GTX 970, GTX 980;
<b>Pascal</b>	GeForce 1000: GTX 1050, 1050 Ti, 1060, 1080;
<b>Turing</b>	GeForce RTX 2000 and GTX 1600: GTX 1660, GTX 1650, RTX 2060, RTX 2080;
<b>Ampere</b>	GeForce RTX 3080, RTX 3090, RTX 3070, etc.

4. Install the latest version of Docker using the command below:

```
curl https://get.docker.com | sh \
&& sudo systemctl --now enable docker
```

**Warning:** If the command doesn't work, please, follow all the steps of [Docker installation on CPU server](#).

5. Setup the **NVIDIA package repository** and the **GPG key**:

```
distribution=$(. /etc/os-release;echo $ID$VERSION_ID) \
&& curl -fsSL https://nvidia.github.io/libnvidia-container/gpgkey | sudo gpg \
--dearmor -o /usr/share/keyrings/nvidia-container-toolkit-keyring.gpg \
&& curl -s -L https://nvidia.github.io/libnvidia-container/$distribution/ \
libnvidia-container.list | \
sed 's#deb https://#deb [signed-by=/usr/share/keyrings/nvidia- \
container-toolkit-keyring.gpg] https://#g' | \
sudo tee /etc/apt/sources.list.d/nvidia-container-toolkit.list
```

6. Update the apt package:

```
sudo apt-get update
```

7. Install the **nvidia-docker2 package** and dependencies:

```
sudo apt-get install -y nvidia-docker2
```

8. Restart the **Docker daemon** to complete the installation after setting the default runtime:

```
sudo systemctl restart docker
```

9. Check the runtime in the config:

```
cat /etc/docker/daemon.json
```

If there is the **nvidia-container-runtime** in the output, installation is successful.

```
> cat /etc/docker/daemon.json
{
  "runtimes": {
    "nvidia": {
      "path": "nvidia-container-runtime",
      "runtimeArgs": []
    }
  }
}
```

Now all requirements for FindFace Lite are followed, please, go to the next [STEP](#) to upload FindFace Lite installer and license on the server.

## 7.2.2 CentOS

1. Check the GPU drivers version using the command below:

```
nvidia-smi
```

**Driver Version:** should be  $\geq 418.81.07$ .

```
> nvidia-smi
Sun Dec 11 18:38:18 2022
```

NVIDIA-SMI 460.80		Driver Version: 460.80		CUDA Version: 11.2	
GPU	Name	Persistence-M	Bus-Id	Disp.A	Volatile Uncorr. ECC
Fan	Temp	Perf	Memory-Usage	GPU-Util	Compute M.
0	GeForce RTX 208...	Off	00000000:01:00.0	On	N/A
43%	45C	P8	668MiB / 11016MiB	24%	Default
		27W / 300W			N/A

```
Processes:
```

GPU	GI	CI	PID	Type	Process name	GPU Memory Usage
ID	ID	ID				
0	N/A	N/A	1966	G	/usr/lib/xorg/Xorg	24MiB
0	N/A	N/A	2218	G	/usr/bin/gnome-shell	83MiB
0	N/A	N/A	2605	G	/usr/lib/xorg/Xorg	403MiB
0	N/A	N/A	13798	G	...474575672989714402,131072	64MiB
0	N/A	N/A	15793	G	...AAAAAAAA= --shared-files	28MiB
0	N/A	N/A	15836	G	alacrity	11MiB
0	N/A	N/A	23349	G	alacrity	11MiB
0	N/A	N/A	25405	G	...RendererForSitePerProcess	3MiB

If it is not so, please, go to the official [NVIDIA guide](#) to install the drivers.

2. Check graphics card model using command below:

```
nvidia-smi -L
```

3. Verify that graphics card has architecture  $\geq$  **Kepler**. Find your graphics cards model in the list below and check:

Architecture (from the oldest to the newest)	Series
<b>Fermi</b>	GeForce 400 and 500: GTX 480, GTX 470, GTX 580, GTX 570;
<b>Kepler</b>	GeForce 600 and 700: Nvidia GTX 680, 670, 660, GTX 780, GTX 770;
<b>Maxwell</b>	GeForce 900: GTX 960, GTX 970, GTX 980;
<b>Pascal</b>	GeForce 1000: GTX 1050, 1050 Ti, 1060, 1080;
<b>Turing</b>	GeForce RTX 2000 and GTX 1600: GTX 1660, GTX 1650, RTX 2060, RTX 2080;
<b>Ampere</b>	GeForce RTX 3080, RTX 3090, RTX 3070, etc.

4. Setup the official **Docker CE** repository:

```
sudo yum-config-manager --add-repo=https://download.docker.com/linux/centos/
↪ docker-ce.repo
```

5. Install the containerd.io package:

```
sudo yum install -y https://download.docker.com/linux/centos/7/x86_64/stable/
↪ Packages/containerd.io-1.4.3-3.1.el7.x86_64.rpm
```

6. Install the latest version of **Docker** using the command below:

```
sudo yum install docker-ce -y
```

7. Ensure the Docker service is running with the following command:

```
sudo systemctl --now enable docker
```

8. Test your Docker installation by running the hello-world container:

```
sudo docker run --rm hello-world
```

The container will be started, you will see the operation success message, then the container will automatically be stopped.

```

~
> docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
2db29710123e: Pull complete
Digest: sha256:faa03e786c97f07ef34423fccceec2398ec8a5759259f94d99078f264e9d7af
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/

```

9. After **Docker** is installed continue with **NVIDIA** installation. Setup the repository and the GPG key using the command below:

```

distribution=$(. /etc/os-release;echo $ID$VERSION_ID) \
  && curl -s -L https://nvidia.github.io/libnvidia-container/$distribution/
  ↳ libnvidia-container.repo | sudo tee /etc/yum.repos.d/nvidia-container-
  ↳ toolkit.repo

```

10. Update the yum package:

```
sudo yum clean expire-cache
```

11. Install the **nvidia-docker2** package and dependencies:

```
sudo yum install -y nvidia-docker2
```

12. Restart the Docker daemon to complete the installation after setting the default runtime:

```
sudo systemctl restart docker
```

13. Check the runtime in the config:

```
cat /etc/docker/daemon.json
```

If there is the **nvidia-container-runtime** in the output, installation is succesful.

```

~
> cat /etc/docker/daemon.json
{
  "runtimes": {
    "nvidia": {
      "path": "nvidia-container-runtime",
      "runtimeArgs": []
    }
  }
}
~

```

Now all requirements for FindFace Lite are followed, please, go to the next *STEP* to upload FindFace Lite installer and license on the server.



## STEP 4. FINDFACE LITE INSTALLER AND THE LICENSE UPLOADING TO THE SERVER

In this step we ask you to put the FindFace Lite installer and the product license on the machine, which you are going to use to work with FindFace Lite. It can be either virtual and physical server or PC meeting *the requirements*.

To upload the installer and the license from the local machine to the virtual one use one of the options: *scp* command or an *SFTP file client*.

**Warning:** It is important to put FindFace Lite installer and the license in the same directory.

### 8.1 Use scp command

Scp allows you to securely copy files between two locations using SSH for encryption.

**Warning:** For using `scp` command you need to have a few things in place:

- SSH installed on both the source and the target machines.
- Root access to both source and target machines.

Scp syntax may be different, it depends on the authentication you use for SSH connection to target machine.

Below you will find descriptions on how:

- *scp* looks like *in general*;
- *to use scp with login and password SSH authentication*;
- *to use scp with secret and public SSH keys authentication*.

### 8.1.1 General scp syntax

```
scp [[user@]src_host:]file1 [[user@]dest_host:]file2
```

Where:

- `scp` initializes the command and ensures a secure shell is in place.
- `src_host` is a source place, where the file is hosted.
- `dest_host` is a target place, where the file will be copied to.

### 8.1.2 Scp syntax for login and password SSH authentication

The format of `scp` command for uploading from the local machine to the server is the following:

```
scp location/file_name.ext username@destination_host:/location
```

Where:

- `scp` initializes the command and ensures a secure shell is in place.
- `location/file_name.ext` is the path to the file, which you want to copy, and its full name.
- `username@destination_host` are credentials for connection: *username* is for a username, *destination\_host* is for a server IP address.
- `/location` is the path where to put the file copy.

---

**Command example:**

```
scp NtechLab_license_512b25bdcd334b44b87ccf5f089215b9.lic azureuser@00.00.000.000:home/azureuser
```

---

After you run the command, you will be asked for a password. Enter it, but note that it won't be displayed.

---

**Command result:**

```
azureuser@00.00.000.000's password:
```

```
NtechLab_license_512b25bdcd334b44b87ccf5f089215b9.lic .....100% 3672KB 126.6KB/s 00:29
```

---

### 8.1.3 Scp syntax for secret and public SSH keys authentication

The format of `scp` command for uploading from the local machine to the server is the following:

```
scp -i ~/.ssh/private_key_name location/file_name.ext username@destination_host:/location
```

Where:

- `scp` initializes the command and ensures a secure shell is in place.
- `~/.ssh/private_key_name` is the path to the private key in `ssh` folder. This part of the command is responsible for the authentication to the server.
- `location/file_name.ext` is the path to the file, which you want to copy, and its full name.

- `username@destination_host` are credentials for connection: *username* is for a username, *destination\_host* is for a server IP address.
- `/location` is the path where to put the file copy.

---

**Command example:**

```
scp -i ~/.ssh/private NtechLab_license_512b25bdcd334b44b87ccf5f089215b9.lic  
azureuser@00.00.000.000:/home/azureuser
```

---

---

**Command result:**

```
azureuser@00.00.000.000:  
NtechLab_license_512b25bdcd334b44b87ccf5f089215b9.lic .....100% 3672KB 126.6KB/s 00:29
```

---

## 8.2 Use an SFTP file manager

1. Install one of the file managers. For example, [FileZilla Client](#) , which is suitable for Windows, Linux and MacOS platforms.
2. Connect to the server with a username and a server IP address.
3. Move the file you need to the server using GUI.



## STEP 5. FINDFACE LITE INSTALLATION

### 9.1 Installation

1. Change the *FindFace Lite installer* file mode to *executive* using the command below:

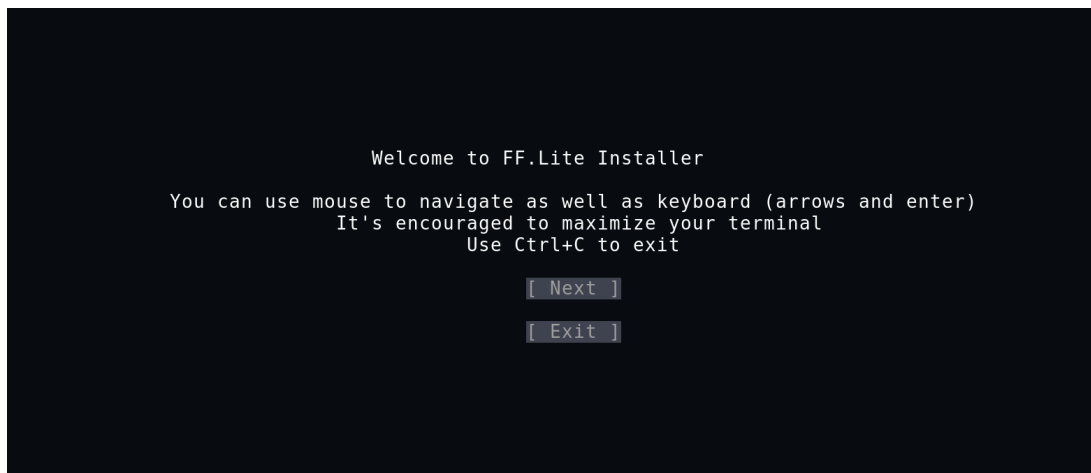
```
chmod +x fflite-{cpu|gpu}-master-g{git_hash}.run
```

Where `fflite-{cpu|gpu}-master-g{git_hash}.run` is the FindFace Lite installer name.

2. Run the installer file:

```
sudo ./fflite-{cpu|gpu}-master-g{git_hash}.run
```

3. Installer interface will be opened within the command line. Press **[Next]**.



4. Wait until the validator checks the software settings and press **[Next]**.

```
Architecture          x86_64
Docker version        19.3.12
Compose version       2.2.3
License               OK

All checks are good

[ Next ]
```

5. After a status check, the installer will start components installation. Wait until installation will be completed and press **Enter**.

```
Loading images
postgres              ok
etcd                  ok
ntls                   ok
vm                     ok
vw                    in progress
eapi                  pending
api                   pending
nginx                 pending

Starting containers
postgres              pending
etcd                  pending
ntls                   pending
vm                     pending
vw                    pending
eapi                  pending
api                   pending
nginx                 pending

Other operations
Running migrations pending
Create admin user pending
```

6. You will see your personal authorization information.

**Warning:** Save the displayed information for future use.

```
UI                      http://172.20.77.
API docs                http://172.20.77. ./api-docs
Installation logs       /tmp/installer_run_217627257.log
Admin username          admin
Admin password          ZlzXnegIUFkB

[ Exit ]
```

7. Installation is finished.

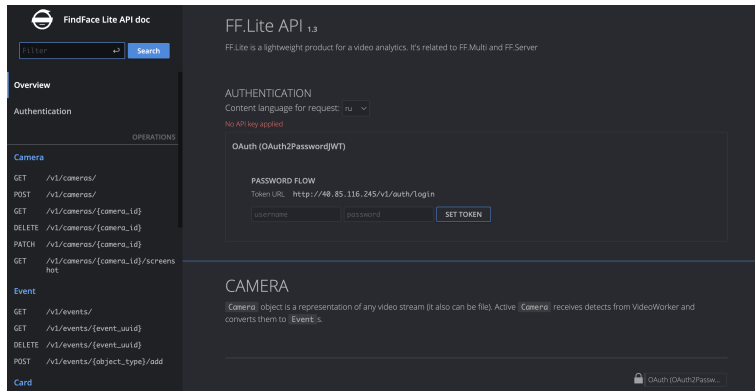
Press **[Exit]** button, the path to the log file of installation will be displayed.

In this log file you can find the credentials to the FindFace Lite.

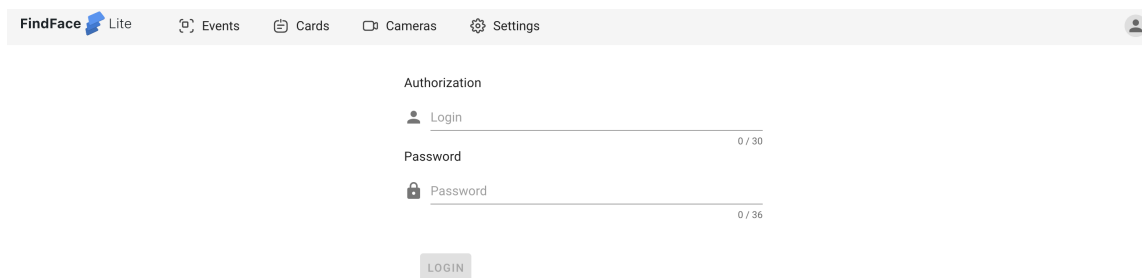
## 9.2 When the installation is finished

Try FindFace Lite service using API and UI:

- API is located on [http://<your\\_hostname>/api-docs](http://<your_hostname>/api-docs). You can also check the description of *API functionality and a guide how to use interactive API documentation*.



- UI URL address is located on [http://<your\\_hostname>](http://<your_hostname>).





## CONFIG SETTINGS

You can fine-tune the FindFace Lite making changes in the configuration file located in the **FFlite** -> **api\_config.yml**. Configuration file includes blocks with information about:

- **app** — API configuration;
- **eapi** — eapi address;
- **eapi\_license\_plate** – license plate eapi address;
- **vm** — vm address and credentials;
- **db** — db address and credentials.

### 10.1 How to work with config settings

Open the file located in the **FFlite** -> **api\_config.yml** using any text editor (*nano*, *vim*, etc.) and make changes in necessary settings.

In sections below you will find the full information about each block and possible settings values.

**Warning:** Please, read the description of the settings below to be sure in the result.

Apply new settings by restarting the **api** service with the following command:

```
docker compose restart api
```

### 10.2 App configuration

#### 10.2.1 Settings possible values

Settings	Possible values
<b>host</b>	<b>0.0.0.0</b> – default value.
<b>port</b>	<b>8000</b> – default value.
<b>debug</b>	<b>false</b> (default) – debug mode is disabled;
	<b>true</b> – debug mode is enabled.
<b>router_base_url</b>	<b>http://nginx</b> – default value.

Table 1 – continued from previous page

Settings	Possible values
<b>media_root</b>	<b>/uploads</b> – default value.
<b>fullframe_root</b>	<b>/fullframe</b> – default value.
<b>normalized_root</b>	<b>/normalized</b> – default value.
<b>save_fullframe</b>	<b>false</b> – fullframe images will not be saved on the disk.
	<b>true</b> (default) — fullframe images will be saved on the disk.
<b>save_normalized</b>	<b>false</b> (default) – normalized images will not be saved on the disk.
	<b>true</b> — normalized images will be saved on the disk.
<b>secret_key</b>	<b>change_me</b> – default value.
<b>max_event_age_days</b>	<b>20</b> – default value.
<b>face_confidence_threshold</b>	<b>0.714</b> – default value.
<b>car_confidence_threshold</b>	<b>0.65</b> – default value.
<b>webhook_workers_num</b>	<b>10</b> – default value.
<b>exit_on_availability_check_fail</b>	<b>false</b> – API service will retry to reach necessary resources until success with exponential timeouts
	<b>true</b> (default) – API service will exit if any of necessary resources are not available.
<b>event_creation_token</b>	<b>change_me</b> – default value.
<b>event_creation_response_type</b>	<b>serialized</b> (default) – the response will contain full information about a created Event, including m
	<b>id</b> – the response will contain only ID of a created Event.
	<b>serialized_verbose</b> – the response will contain full information about a created Event (including a
<b>face_features</b>	<b>headpose</b> – position of a head;
	<b>medmask</b> – detection of a medmask;
	<b>liveness</b> – liveness detection.
<b>car_features</b>	<b>orientation</b> – recognition a car position while recognition.
	<b>special_types</b> – recognition of a car type.
	<b>license_plate_visibility</b> – recognition of license plate.
<b>liveness_source</b>	<b>eapi</b> – liveness detection from image. This value is required to be set if you want to create Events
	<b>vw</b> – liveness detection from videostream.
<b>auth_enabled</b>	<b>false</b> – authorization is disabled;
	<b>true</b> (default) – authorization is enabled.
<b>access_token_expire_minutes</b>	<b>43200</b> – default value.
<b>dedup_enabled</b>	<b>false</b> – deduplication of Events is disabled;
	<b>true</b> (default) – deduplication of Event is enabled.
<b>save_dedup_events</b>	<b>false</b> (default) – Events duplicates will be saved.
	<b>true</b> – Events duplicates will not be saved.
<b>face_dedup_interval</b>	<b>5</b> – default value.
<b>face_dedup_confidence</b>	<b>0.9</b> – default value.
<b>car_dedup_interval</b>	<b>5</b> – default value.
<b>car_dedup_confidence</b>	<b>0.9</b> – default value.

## 10.2.2 Settings description

Settings	Description
<b>host</b>	Host information
<b>port</b>	Port information
<b>debug</b>	Debug mode. Currently manage <b>only</b> debug logs.
<b>router_base_url</b>	Router base URL for VM. Please, change only if you are sure.
<b>media_root</b>	Root directory for media files, which stores the <i>Objects</i> images.
<b>fullframe_root</b>	Root directory for fullframe files, which stores frame images from VideoWorker (vw).
<b>normal- ized_root</b>	Root directory for normalized files, which stores files used for migrations between models in database.
<b>save_fullframe</b>	Saving settings of fullframe images.
<b>save_normalized</b>	Saving settings of normalized images.
<b>secret_key</b>	A secret key, needed for security operations.
<b>max_event_age_days</b>	Maximum time, which Event is stored. After the expiration of the set time period, the Event is deleted.
<b>face_confidence_threshold</b>	Face confidence threshold, according to which an Event matches or not with a Card during the face matching. If matching score is more than set value, Event is matched with a Card.
<b>car_confidence_threshold</b>	Car confidence threshold, according to which an Event matches or not with a Card during the car matching. If matching score is more than set value, Event is matched with a Card.
<b>web-hook_workers_num</b>	Number of concurrent webhook workers sending requests to webhook targets.
<b>exit_on_availability_check_fail</b>	Exit API service in case of unavailability of necessary resources.
<b>event_creation_token</b>	Token using for external detector authentication for Event creation (/ {object_type}/add). JWT token doesn't apply in this request.
<b>event_creation_response_type</b>	Response type of Event creation (/ {object_type}/add) request.
<b>face_features</b>	Features of FindFace Lite for face recognition. A feature is activated if it is listed in the value. After feature activation freshly created events will be populated with corresponding feature.
<b>car_features</b>	Features of FindFace Lite for car recognition. A feature is activated if it is listed in the value. After feature activation freshly created events will be populated with corresponding feature.
<b>liveness_source</b>	Source of liveness detection.
<b>auth_enabled</b>	Authorization managing. Note, that all API calls (with some exceptions) will require <i>Autho- rization</i> header with <i>JWT &lt;token&gt;</i> .
<b>ac- cess_token_expire_minutes</b>	Time of access token expiration interval.
<b>dedup_enabled</b>	Events deduplication managing.
<b>save_dedup_event</b>	Events duplicates saving settings.
<b>face_dedup_interval</b>	Time interval in seconds during which Events with the same person will be considered as duplicates.
<b>face_dedup_confidence</b>	Confidence of matching between 2 Events to be considered duplicates.
<b>car_dedup_interval</b>	Time interval in seconds during which Events with the same car will be considered as duplicates.
<b>car_dedup_confidence</b>	Confidence of matching between 2 Events to be considered duplicates.

## 10.3 EAPI configuration

Settings	Possible values	Description
host	eapi – default value.	Host information
port	18666 – default value.	Port information

## 10.4 License plate EAPI configuration

Settings	Possible values	Description
host	eapi-license-plate – default value.	Host information
port	18667 – default value.	Port information

## 10.5 VM configuration

Set-tings	Possible values	Description
host	vm – default value.	Host information
port	18810 – default value.	Port information
token	GOOD_TOKEN – default value.	Token for processes connected with vm. Should be the same as token in <i>vm.conf</i> file.

## 10.6 DB configuration

Settings	Possible values	Description
host	postgres – default value.	Host information
port	5432 – default value.	Port information
user	fflite – default value.	Credentials to access database with the name from <b>database</b> setting.
password	fflite – default value.	
database	fflite – default value.	Database name.

## API

FindFace Lite API is located on `http://<your_hostname>/api-docs`. It is interactive, which means that you can make requests and get responses right on this page.

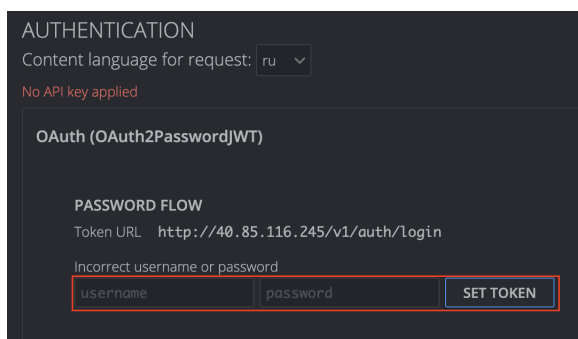
API documentation allows to read, create, update and delete all entities and provides description for all methods and parameters.

In this article we will overview the blocks of FindFace Lite functionality accessible via API and how to use interactive API.

### 11.1 Preparation to API usage

Before use the FindFace Lite API, please, authenticate yourself by creating JWT token in the **AUTHENTICATION** section.

Enter **username** and **password** from the *STEP 5* of **Getting started** block into the form and click the **SET TOKEN** button.



AUTHENTICATION

Content language for request: ru

No API key applied

OAuth (OAuth2PasswordJWT)

PASSWORD FLOW

Token URL http://40.85.116.245/v1/auth/login

Incorrect username or password

username password SET TOKEN

After authentication you can use interactive FindFace Lite API.

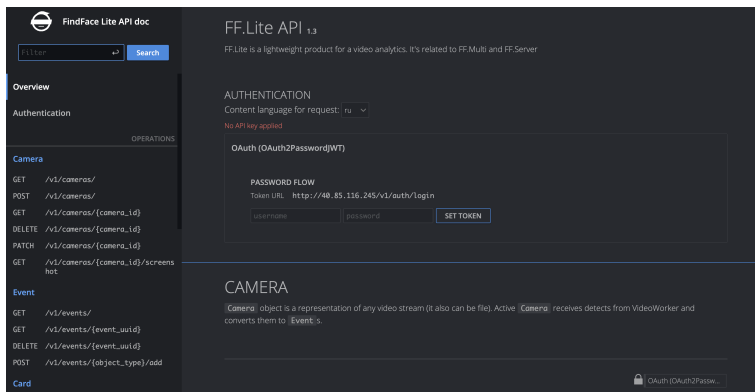
---

**Note:** If you need to use API requests outside the interactive API, please, use the created token.

---

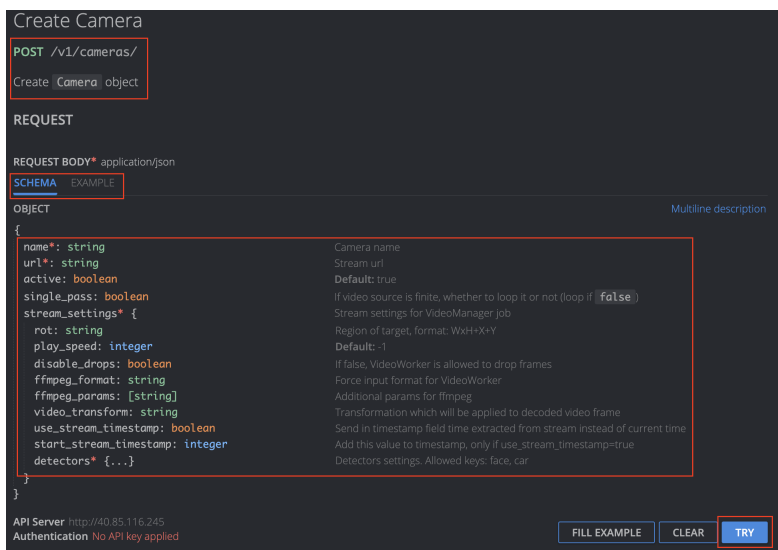
## 11.2 API usage overview

API page is divided into 2 parts: left is a list of operations and right are operations execution field.

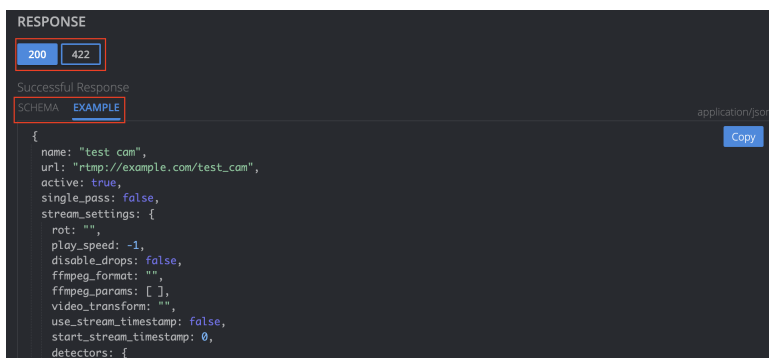


Each operation consists of **Request** and **Response** part:

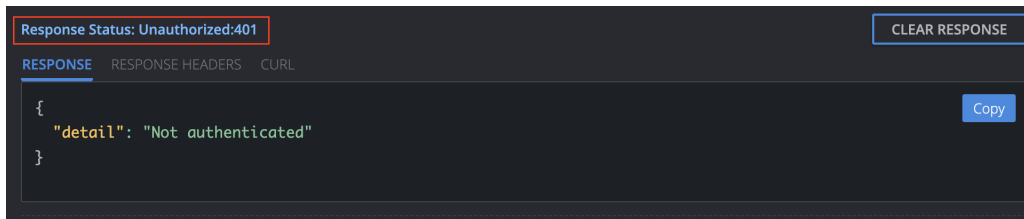
- Request part describes the operation, including **request schema** with the interpretation of each parameter, **request example** and **TRY** button, which sends the request.



- Response part describes the **schema** and **examples** of each variant of response on the given operation.



After the **TRY** button is clicked, request is sent and you will see the response block with the **response status** and detailed information.

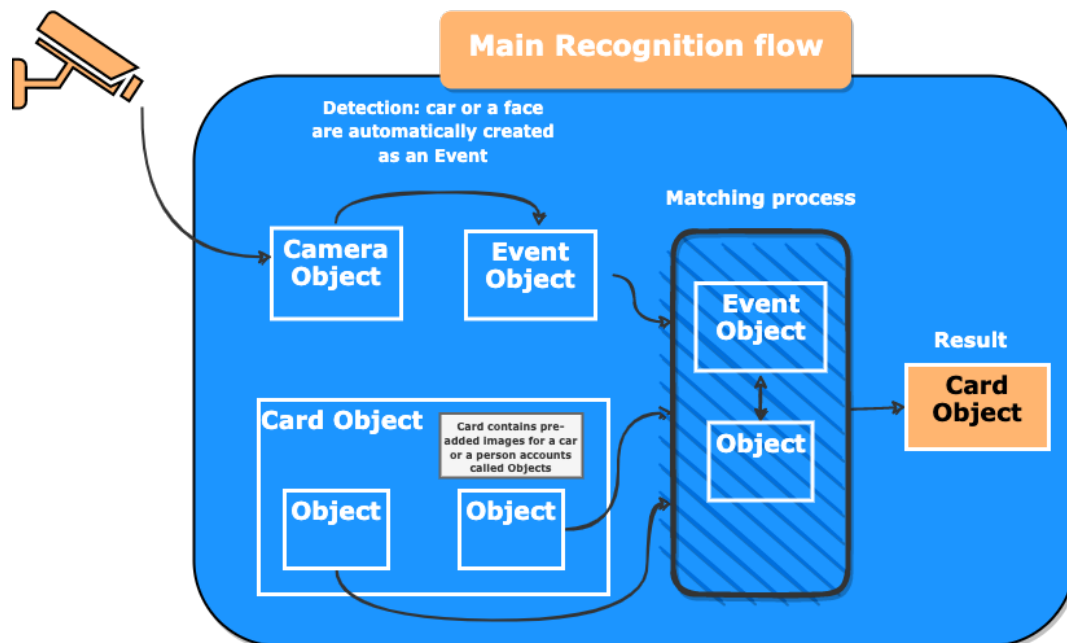


## 11.3 FindFace Lite API functionality

API documentation can be divided into the semantic blocks, which contain all requests (to get, add, edit and delete entities):

### 1. Recognition flow

- **Camera** block manages Camera object, which is a representation of any video stream (it also can be a file). Active Camera receives detection data from *VideoWorker* and converts it to Events.
- **Event** block manages Event object, which is a representation of an object (face or car) occurrence in the camera frame. With active Camera Event is automatically created from VideoWorker detection data. You or any 3rd party system can also create it outside the main flow using POST request.
- **Card** block manages Card object, which is a profile of a real person or a car. Card can be one of two types: *face* or *car*.
- **Object** block manages Object, which is a representation of particular face or car. To create it, you have to add the image and link it to the Card.



### 2. External system intereaction

- **Webhook** block can be used to notify external systems about Events and matches.

### 3. Authentification and user management

- **Auth** block describes methods used for authentication.
- **User** block manages Findface Lite users

#### 4. System operations

- **Misc** block contains undefined requests, connected with the service needs.
- **Pipeline** block is an internal method for *VideoWorker* and normally is not used in usual flow.

## EDGE DEVICES

### 12.1 What are edge devices

Edge devices are physical devices (e.g., *Identity authentication terminals*), which can connect and send images to FindFace Lite to recognise objects. In order to these devices get the results of recognition *webhooks* should be set.

The process of recognition from edge devices is the same as from CCTV cameras, the only difference is that Events are created directly from an edge device via POST request, but not from FindFace Lite system.

### 12.2 Preparation to recognition process

Before edge device integration, authenticate a device and create *Objects* and *Cards*, which will be compared with the created *Events*.

#### 12.2.1 Authenticate a device

In order to execute all operations (except those which are connected with Events) authenticate a device in the system, using **/v1/auth/login** POST request. For Events there is a token from the *config file* is used.

For **username** and **password** parameters use data you got on *the STEP 5*.

**Request example:**

```
{
  "username": "login",
  "password": "password"
}
```

**Successful response example:**

```
{
  "access_token": "token"
}
```

### 12.2.2 Create a Card

*Card* is used for keeping several Objects of a person or a car under the one profile. In the process of recognition the Card is treated as a *result*.

To create a Card, use the `/v1/cards/` POST request. All parameters are described below.

Parameters	Value types	Description
<b>name</b>	string	The name of a Card.
<b>active</b>	true	The Card is enabled.
	false	The Card is disabled.
<b>type</b>	face	The Card is created for the face recognition.
	car	The Card is created for the car recognition.
<b>wiegand</b>	string	Wiegand code.

Request example:

```
{
  "name": "test card",
  "active": true,
  "type": "face",
  "wiegand": "test wiegand code"
}
```

Successful response example:

```
{
  "name": "test card2",
  "active": true,
  "type": "face",
  "wiegand": "test wiegand cod2e",
  "id": 2,
  "objects": []
}
```

### 12.2.3 Create an Object

*Objects* are used for representation of a face or a car. To create it add the image and link to the Card, using `/v1/objects/` POST request.

All operations are described below.

Parameters	Value types	Description
<b>card_id</b>	a number	The Card ID, which you want to connect with the Object. With one Card can be connected several Objects.
<b>type</b>	enum	The type of the Object you want to create. <b>ar</b> is for car image, <b>face</b> is for face image and <b>license_plate</b> is for license plate image.
<b>in-put_file</b>	string	Put the filename which contains a face or car you would like to add to the database of Objects.

Request example:

```
{
  "card_id": "2",
  "type": "face",
  "input_file": "somehash.jpg"
}
```

Successful response example:

```
{
  "id": 1,
  "emben": "vV1yPfc2izy...de8vY/bvNXLfDw=",
  "type": "face",
  "card": 4,
  "filename": "somehash.jpg"
}
```

## 12.3 Edge Devices integration

To integrate an edge device with FindFace Lite, use API. All operations are described below.

### 12.3.1 Create a Camera

Create *Camera* object using `/v1/cameras/` POST request.

**Note:** Camera object is only needed for further Event objects creations. It is not used in the flow of recognition via edge devices.

Here is the description of needed parameters. Parameters, which are not included, describe settings for streams. As you do not have any streams, you may not pay attention to them.

Parameters	Description
<b>name</b>	The name for the Camera object. You can choose any.
<b>url</b>	URL of an added stream. As you don't have the stream, set any value, beginning with <code>rtmp://</code> .
<b>active</b>	Camera object status. Set it to disabled – <code>false</code> .
<b>stream_settings</b>	Settings for streams from the CCTV camera. You have to fill only the mandatory parameter – <b>detectors</b> .
<b>detectors</b>	Detectors settings. You have to include this parameter into the request, but you can leave it empty.

Request example:

```
{
  "name": "Edge device",
  "url": "rtmp://none",
  "active": false,
  "stream_settings": {
    "detectors": {
    }
  }
}
```

Successful response example:

```
{
  "name": "Edge device",
  "url": "rtmp://none",
  "active": false,
  "single_pass": false,
  "stream_settings": {
    "rot": "",
    "play_speed": -1,
    "disable_drops": false,
    "ffmpeg_format": "",
    "ffmpeg_params": [],
    "video_transform": "",
    "use_stream_timestamp": false,
    "start_stream_timestamp": 0,
    "detectors": {}
  },
  "id": 2,
  "status": "UNKNOWN"
}
```

### 12.3.2 Configure the Edge Device

**Warning:** After this step the edge device will be able to send data to FindFace Lite for recognition. To get the results back you should set the [Webhook](#).

Configure the edge device to send Events `/v1/events/{object_type}/add` POST requests. Event in case of edge device is a representation of an object (face or car) occurrence in the edge device zone, which is sent as a static file to the FinFace lite via API.

Here is the description of parameters the edge device should send to FindFace Lite API.

Parameters	Value types	Description
object_type	car	The path parameter /v1/events/{object_type}/add, which specify the object of recognition.
	face	
	license_plate	
token	string	Authorization via <b>event_creation_token</b> set in <i>configuration file</i> .
camera	a number	Camera ID with which the Event will be connected.
fullframe	binary	Image in any suitable for static content format (jpeg, png, etc.).
rotate	true	The technology of image rotation is enabled. System checks the objects position and tries to rotate if objects are upside down.
	false	The technology of image rotation is disabled.
timestamp	date-time	The data-time in ISO format <b>yyyy-MM-dd'T'HH:mm:ss.SSSXXX</b> , for example: <b>2000-10-31T01:30:00.000-05:00</b>
mf_selector	all	Multiface selector is enabled. All objects of Event are detected.
	biggest	Multiface selector is disabled. There is only biggest object on the image is detected.
roi	numbers	Region of interest, which means image detection area. Specify the value in the [left, top, right, bottom] format, where values in the brackets are the numbers in grades

**Request example:**

```
{
  "object_type": "face",
  "token": "change_me",
  "camera": 2,
  "fullframe": "somehash.jpg",
  "rotate": true,
  "timestamp": "2000-10-31T01:30:00.000-05:00",
  "mf_selector": "biggest",
  "roi": 15,20,12,14
}
```

**Successful response example:**

The response view will be different, depending on the value of the setting `event_creation_response_type` in the *config file*.

- If ID is set, you will see only main information and the ID of a created Event

```
{
  "events": [
    "cc04cc9c-f355-4121-80c4-94a02eec652a",
    "c7d51db3-5b52-4318-9565-e2651308c1a6"
  ]
}
```

- If `serialized` is set, you will see full information about a created Event, including matched Card, path to the fullframe image, etc.

```
{
  "events": [
```

(continues on next page)

(continued from previous page)

```

{
  "bbox_bottom": 97,
  "bbox_left": 170,
  "bbox_right": 214,
  "bbox_top": 39,
  "bs_type": "realtime",
  "camera": 1,
  "card": null,
  "confidence": null,
  "created_date": "2022-12-29 13:02:07.910724+00:00",
  "emben":
    ↪ "bmY3Pff9Grt1Ah09lp8kvn+a6Tw8SZs8K5xtvL0jtrxFtJ+9d5WIPH3PHL39acg9oNWhu4Mv2j2VjPo8QqDjubiFkz05BouS
    ↪ aQBvfdqFD2/
    ↪ woc9j03i05U3vT1P6ya9BfNyPUCBkz1Smmm8CIPvPRPxWTzWXxo8DwRGvMxfr7zRhGw8KyZzPtwoCT0Bx7C9AcKWvflgUb2NI
    ↪ BD70U8M9grEbPVTZCb3mgXg+LxfEvd6uL2wLh08BU6yPQhREz0kj1M98tY+PNbA9D1MDC07Tp4dPjh7n7zZAQS9/
    ↪ JFzPWEJCb1CTwU+3deFvQb/hz2YaAa+Qbjpvd0UFb7HVtG9NEhLPMfS1rouTk49f6DT0y9r/
    ↪ j373aG7hBa3PW/
    ↪ eJT5Zuz08c0+2PZXVDL3hemE8sYa0PY7Xtj06NAG+Stw4vrhQhb2KtJe9J8hCvTM/
    ↪ Gr4vr0m9S3jJvfnBH2QVgk97HuzvH2V/
    ↪ 71lQsk9l4UTvqE3XD1ssos9ErvcvY5CCr6ftJ+9EOarP0KvKr0JFa69sdyCvQPcu2nLt089I+iuzS+aj1bTgY+qT4dPtREVz
    ↪ rVPA==",
  "features": {
    "headpose": {
      "pitch": 11.207756,
      "roll": 3.1429868,
      "yaw": 4.652887
    },
    "liveness": 0.6584981
  },
  "fullframe": "2022/12/29/13/c436c2d92c4c627d5c6d13f9f1d9555a.jpg",
  "type": "face",
  "uuid": "2f25dd19-d0cd-4b44-9147-69a7dc57450e"
},
{
  "bbox_bottom": 75,
  "bbox_left": 57,
  "bbox_right": 100,
  "bbox_top": 15,
  "bs_type": "realtime",
  "camera": 1,
  "card": 1,
  "confidence": 0.8306973576545715,
  "created_date": "2022-12-29 13:02:07.917896+00:00",
  "emben": "yRukvS0o1r2Bcm699SGT0/
    ↪ SPBz64MvK9xINqPYDge708nSW9ba4BvAVgZ77ctw7A20hvZ5LRD2rA1488DLnvSKQXT0ER367zf20vdypqb2Lhog8nIxjPaZ
    ↪ wKlvVlwr724Ll09LdqYPeOq170bDua8Zn2lPNy8dr0TF9G7VhMXPT6yQz0aCI49h80JPQVTKb25oB+9x9++PEKCFj4uq0i8bF
    ↪ PR3mrbsx+rW7TQqNO4QcFz2oI407H4nfvY/
    ↪ nQD57Y2m9ItFMPZQKibzobRi6cf9wPT1itz0lkW89qUv0vS9RVDxjGoC9E3Si0xqSsbzjnyc+P4ZnPpFjEz5XMZE8IuILvYvg
    ↪ A9waic0x90n7z0kW+9k7EmvIxxuLwdPPo9t5H3vQETLz6FcGQ+1f0qvfkWkZ0rfS09ckoivV65k70xw5296raIvYnnE70gaYa
    ↪ ",
  "features": {
    "headpose": {

```

(continues on next page)

(continued from previous page)

```

        "pitch": -2.1808214,
        "roll": -0.5856089,
        "yaw": -4.5041146
    },
    "liveness": 0.5574532
},
"fullframe": "2022/12/29/13/4ef96c620d738d87c00aaaaa12fccca2.jpg",
"type": "face",
"uuid": "9b718d45-919a-490f-9fe6-b2af58cbf83a"
}
]
}

```

- If `serialized_verbose` is set, you will see full information about a created Event (including a matched Card, path to the fullframe image, etc) and full information about the Card.

```

{
  "events": [
    {
      "bbox_bottom": 97,
      "bbox_left": 170,
      "bbox_right": 214,
      "bbox_top": 39,
      "bs_type": "realtime",
      "camera": {
        "active": false,
        "id": 1,
        "name": "test camera",
        "single_pass": false,
        "status": "DISABLED",
        "url": "rtmp://test"
      },
      "card": null,
      "confidence": null,
      "created_date": "2022-12-29 13:48:34.624541+00:00",
      "emben":
        ↪ "bmY3Pff9Grt1Ah09lp8kvn+a6Tw8SZs8K5xtvL0jtrxFtJ+9d5WIPH3PHL39acg9oNWhu4Mv2j2VjPo8QqDjubiFkz05BouS
        ↪ aQBvfdqFD2/
        ↪ woc9j03i05U3vT1P6ya9BfNyPUCBkz1Smm8CIPvPRPxWTzWXxo8DwRGvMxfr7zRhGw8KyZzPtwoCT0Bx7C9AcKWvflgUb2NL
        ↪ BD70U8M9grEbPVTZCb3mgXg+LxfEvd6uL2wLh08BU6yPQhREz0kj1M98tY+PNbA9D1MDC07Tp4dPjh7n7zZAQS9/
        ↪ JFzPWEJCb1CTwU+3deFvQb/hz2YaAa+Qbjpvd0UFb7HVtG9NEhLPMfS1rouTk49f6DT0y9r/
        ↪ j373aG7hBa3PW/
        ↪ eJT5Zuz08c0+2PZXVDL3hemE8sYa0PY7Xtj06NAG+Stw4vrhQhb2KtJe9J8hCvTM/
        ↪ Gr4vr0m9S3jJvfnBH22QVgk97HuzvH2V/
        ↪ 71lQsk9l4UTvqE3XD1ssos9ErvY5CCr6ftJ+9EOarPOKvKr0JFa69sdyCvQPcuJ2nLt089I+iuzS+aj1bTgY+qT4dPtREVz
        ↪ rVPA==",
      "features": {
        "headpose": {
          "pitch": 11.207756,
          "roll": 3.1429868,
          "yaw": 4.652887
        }
      },
    }
  ],
}

```

(continues on next page)

(continued from previous page)

```

        "liveness": 0.6584981
    },
    "fullframe": "2022/12/29/13/5e870f4f9dbd1e27652f6384663b8cab.jpg",
    "type": "face",
    "uuid": "df0821b4-6e52-4b66-abd2-0f642e2a090a"
},
{
    "bbox_bottom": 75,
    "bbox_left": 57,
    "bbox_right": 100,
    "bbox_top": 15,
    "bs_type": "realtime",
    "camera": {
        "active": false,
        "id": 1,
        "name": "test camera",
        "single_pass": false,
        "status": "DISABLED",
        "url": "rtmp://test"
    },
    "card": {
        "active": true,
        "id": 1,
        "name": "test card",
        "objects": [],
        "type": "face",
        "wiegand": "test wiegand code"
    },
    "confidence": 0.8306973576545715,
    "created_date": "2022-12-29 13:48:34.633562+00:00",
    "emben": "yRukvS0o1r2Bcm699SGT0/
    ↪ SPBz64MvK9xINqPYDge708nSW9ba4BvAVgLz77ctw7A20hvZ5LRD2rA1488DLnvSKQXT0ER367zf20vdypqb2Lhog8nIxjPaZ
    ↪ wKlvVlwr724Ll09LdqYPeOq170bDua8Zn2lPNy8dr0TF9G7VhMXPT6yQz0aCI49h80JPQVTKb25oB+9x9++PEKCFj4uq0i8bE
    ↪ PR3mrbxs+rW7TQqNO4QcFz2oI407H4nfvY/
    ↪ nQD57Y2m9ItFMPZQKibzobRi6cf9wPT1itz0lkW89qUv0vS9RVDxjGoC9E3Si0xqSsbzjnyc+P4ZnPpFjEz5XMZE8IuILvYvg
    ↪ A9waic0x90n7z0kW+9k7EmvIxxuLwdPPo9t5H3vQETLz6FcGQ+1f0qvfkWkZ0rfS09ckoivV65k70xw5296raIvYnnE70gaYa
    ↪ ",
    "features": {
        "headpose": {
            "pitch": -2.1808214,
            "roll": -0.5856089,
            "yaw": -4.5041146
        },
        "liveness": 0.5574532
    },
    "fullframe": "2022/12/29/13/6041c2a71f4e2020d4cbaa52ce9b41f8.jpg",
    "type": "face",
    "uuid": "4dad4c16-f1cd-4ff1-a18a-268b71c1dbec"
}
]
}

```

## WEBHOOKS

*Webhook* is a user-defined HTTP callbacks, triggering by an event in a web app.

You can use webhooks for various purposes, for instance, to notify a user about a specific Event, invoke required behavior on a target website, and solve security tasks such as automated access control.

For example, if you set up an edge device and want to send the result of recognition back to proceed with the object verification.

In order to FindFace Lite send an HTTP request to the URL when a required event occurs, configure the webhook.

### 13.1 Authenticate a device

To authenticate your device in the system, use **/v1/auth/login** POST request.

For **username** and **password** parameters use data you got on *the STEP 5*.

**Request example:**

```
{
  "username": "login",
  "password": "password"
}
```

**Successfull response example:**

```
{
  "access_token": "token"
}
```

### 13.2 Create a Webhook

To create a webhook use **/v1/webhooks/** POST request. Below are described all parameters and values type:

Parameters	Value types	Description
<b>name</b>	string	The name of a webhook.
<b>active</b>	true	The default value, which means that the webhook is enabled.
	false	The possible value, which means that the webhook is disabled.
<b>target</b>	string	Target URL to call when an event happens.

Table 1 – continued from pre

Parameters	Value types	Description
<b>filters</b>	object [filters]	A set of filters, according to which messages are sent or not to the tar
<b>type_in</b>	string	If the created Event is matched with the specified types (face, car or l
<b>camera_in</b>	a number or several numbers	If the created Event is connected with the specified Camera ID, a me
<b>card_in</b>	a number or several numbers	If the created Event is matched with the specified Card ID, a message
<b>confidence_gte</b>	a number from 0 to 1	If the result of recognition is greater than or equal to the specified va
<b>confidence_lte</b>	a number from 0 to 1	If the result of recognition is less than or equal to the specified value,
<b>matched</b>	true	Only matched Events will trigger a message sending to the target UR
	false	Only unmatched Events will trigger a message sending to the target U
<b>bs_type_in</b>	overall	Only the best result of recognition for a particular time will trigger a
	realtime	All results of recognition will trigger a message sending to the target
<b>Yaw, Pitch, Roll</b>		Parameters mean the angle of rotation in grades. They can only be ap
<b>yaw_lte</b>	a number	If the headpose yaw is less than or equal to the specified value, a mes
<b>yaw_gte</b>	a number	If the headpose yaw is greater than or equal to the specified value, a m
<b>pitch_lte</b>	a number	If the headpose pitch is less than or equal to the specified value, a me
<b>pitch_gte</b>	a number	If the headpose pitch is greater than or equal to the specified value, a
<b>roll_lte</b>	a number	If the headpose roll is less than or equal to the specified value, a mess
<b>roll_gte</b>	a number	If the headpose roll is greater than or equal to the specified value, a n
<b>liveness_lte</b>	a number from 0 to 1	If the liveness level is less than or equal to the specified value, a mess
<b>liveness_gte</b>	a number from 0 to 1	If the liveness level is greater than or equal to the specified value, a m
<b>medmask</b>	object	If the <b>medmask</b> feature is specified in the configuration file, you ca
<b>name</b>	enum	Available options according to which events can be filtered:
		<b>none</b> – there is no medmask on the face;
		<b>correct</b> – medmask is put on correctly;
		<b>incorrect</b> – medmask is put on incorrectly.
<b>confidence_lte</b>	a number from 0 to 1	If the confidence value of medmask recognition result is less than or
<b>confidence_gte</b>	a number from 0 to 1	If the confidence value of medmask recognition result is greater than
<b>orientation</b>	object	If the <b>orientation</b> feature is specified in the configuration file, you c
<b>name</b>	enum	Available event recognition results according to which they can be fil
		<b>back</b> – vehicle rear part;
		<b>side</b> – vehicle side part;
		<b>front</b> – vehicle front part.
<b>confidence_lte</b>	a number from 0 to 1	If the confidence value of car orientation recognition result is less tha
<b>confidence_gte</b>	a number from 0 to 1	If the confidence value of car orientation recognition result is less tha
<b>special_type</b>	object	If the <b>special_type</b> feature is specified in the configuration file, you
<b>name</b>	enum	Available event recognition results according to which they can be fil
		<b>not_special</b> – ordinary vehicle,
		<b>police</b> ,
		<b>ambulance</b> ,
		<b>road_service</b> ,
		<b>gas_service</b> ,
		<b>rescue_service</b> ,
		<b>other_special</b> – all other special vehicles that are not specified as a s
		<b>taxi</b> ,
		<b>route_transport</b> ,
		<b>car_sharing</b> ,
		<b>military</b> .
<b>confidence_lte</b>	a number from 0 to 1	If the confidence value of car type recognition result is less than or e

Table 1 – continued from pre

Parameters	Value types	Description
<b>confidence_gte</b>	a number from 0 to 1	If the confidence value of car type recognition result is greater than o
<b>license_plate_number</b>	string	If the recognised car license plate number is equal to the specified va
<b>license_plate_visibility</b>	object	<b>license_plate_visibility</b> is mandatory feature and enabled implicitly
<b>name</b>	enum	Available event recognition results according to which they can be fil
		<b>partly_visible_no_text</b> – license plate is without text and partly visib
		<b>fully_visible_no_text</b> – license plate is without text and fully visible
		<b>invisible</b> – license plate is invisible,
		<b>partly_visible</b> – license plate is with text and partly visible,
		<b>fully_visible</b> – license plate is with text and fully visible.
<b>confidence_lte</b>	a number from 0 to 1	If the confidence value of car orientation recognition result is less tha
<b>confidence_gte</b>	a number from 0 to 1	If the confidence value of car orientation recognition result is less tha
<b>license_plate_event_number</b>	string	Filter can only be applied to <b>license_plate</b> Event type. If the recogni
<b>send_attempts</b>	a number	Numbers of attempts to send a message to the target URL. The attem

**Request example:**

```
{
  "name": "test webhook",
  "active": true,
  "target": "http://localhost/webhok_test",
  "filters": {
    "camera_in": [
      1,
      2
    ],
    "card_in": [
      4,
      5
    ],
    "confidence_gte": 0.75,
    "confidence_lte": 0.79,
    "matched": true,
    "bs_type_in": [
      "overall",
      "realtime"
    ],
    "yaw_lte": 3.5,
    "yaw_gte": 3.5,
    "pitch_lte": -4.2,
    "pitch_gte": -4.2,
    "roll_lte": 1.8,
    "roll_gte": 1.8,
    "liveness_lte": 0.44,
    "liveness_gte": 0.44
  },
  "send_attempts": 3
}
```

**Successfull response example:**

```
{
```

(continues on next page)

(continued from previous page)

```
"name": "test webhook",
"active": true,
"target": "http://localhost/webhook_test",
"filters": {
  "camera_in": [
    1,
    2
  ],
  "card_in": [
    4,
    5
  ],
  "confidence_gte": 0.75,
  "confidence_lte": 0.79,
  "matched": true,
  "bs_type_in": [
    "overall",
    "realtime"
  ],
  "yaw_lte": 3.5,
  "yaw_gte": 3.5,
  "pitch_lte": -4.2,
  "pitch_gte": -4.2,
  "roll_lte": 1.8,
  "roll_gte": 1.8,
  "liveness_lte": 0.44,
  "liveness_gte": 0.44
},
"send_attempts": 3,
"id": 1
}
```

## INDEX

### A

Antispam events, [3](#)

### C

Camera, [3](#)

Card, [3](#)

### D

Deduplication, [3](#)

### E

Edge device, [4](#)

Event, [3](#)

### F

FindFace Lite, [3](#)

FindFace Lite Installer, [3](#)

### I

Identity authentication terminal, [3](#)

### L

Liveness detection, [3](#)

### O

Object, [3](#)

### P

PACS, [3](#)

### V

VideoWorker, [3](#)

### W

Webhook, [3](#)