

# Configuration Guide

## Contents

|  |           |
|--|-----------|
| <b>Configuration Guide</b>                           | <b>3</b>  |
| Configuration file location . . . . .                | 3         |
| <b>Settings</b>                                      | <b>5</b>  |
| System settings . . . . .                            | 5         |
| Descriptor factory settings . . . . .                | 6         |
| FaceDetV3 detector settings . . . . .                | 8         |
| FaceDetV1 detector settings . . . . .                | 10        |
| FaceDetV2 detector settings . . . . .                | 11        |
| LNet . . . . .                                       | 12        |
| LNetIR . . . . .                                     | 12        |
| SLNet . . . . .                                      | 12        |
| HumanDetector settings . . . . .                     | 13        |
| IndexBuilder settings . . . . .                      | 14        |
| Quality estimator settings . . . . .                 | 15        |
| HeadPoseEstimator settings . . . . .                 | 16        |
| EyeEstimator settings . . . . .                      | 17        |
| AttributeEstimator settings . . . . .                | 18        |
| GlassesEstimator settings . . . . .                  | 19        |
| OverlapEstimator settings . . . . .                  | 20        |
| ChildEstimator settings . . . . .                    | 21        |
| LivenessIREstimator settings . . . . .               | 22        |
| HeadAndShouldersLivenessEstimator settings . . . . . | 23        |
| Mouth Estimator settings . . . . .                   | 24        |
| Medical mask estimator settings . . . . .            | 25        |
| Depth Estimator settings . . . . .                   | 27        |
| LivenessFlyingFaces Estimator settings . . . . .     | 28        |
| LivenessRGBM Estimator settings . . . . .            | 29        |
| LivenessOneShotRGBEstimator . . . . .                | 30        |
| Credibility Estimator settings . . . . .             | 31        |
| <b>Runtime settings</b>                              | <b>32</b> |

## Configuration Guide

Configuration options are specified via `faceengine.conf` file which is basically an XML document with special tag formatting. The document itself is not required to exist, in this case FSDK will fall back to some default settings, which, however, may not be suitable for several tasks.

**WARNING!** By changing any configuration settings from default ones it is assumed that user understands what these settings do and how they will affect performance and output results of their application. The rule of thumb is this: DO NOT change anything in configuration file unless you really have to.

Always remember that incorrect config may huck the things up very badly. Pay attention to what you configure and how. Always double-check what you deploy.

Some configuration settings may be omitted due to their obscurity and research use case only.

The location where the config file is found varies across different systems but tries to be as consistent as possible. Path resolution is the following:

### Windows

- Look for `data/faceengine.conf` in current working directory

### Linux

- Look for `/etc/visionlabs/faceengine.conf`
- Look for `data/faceengine.conf` in current working directory if previous options fail

The config file format is optimized for deserialization of several FSDK types:

- Int1 - scalar 32 bit integral numeric type
- Int2 - 2-d 32 bit integral numeric type (aka Vector2i, Size)
- Int3 - 3-d 32 bit integral numeric type
- Int4 - 4-d 32 bit integral numeric type (aka Rect)
- Float1 - scalar 32 bit floating point numeric type
- Float2 - 2-d 32 bit floating point numeric type (aka Vector2f)
- Float3 - 3-d 32 bit floating point numeric type
- Float4 - 4-d 32 bit floating point numeric type
- String - short null-terminated string (max. 16 characters including the null-terminator)

## Configuration file location

The location where the config file is found varies across different systems but tries to be as consistent as possible. Path resolution is the following:

### Windows:

- Look for `data/faceengine.conf` in current working directory

## Linux

- Look for `/etc/visionlabs/faceengine.conf`
- Look for `data/faceengine.conf` in current working directory if previous options fail

## Mobile platforms

- Look for “`data/faceengine.conf`” in current working directory.

## Settings

### System settings

| Parameter           | Description  | Type            | Default value |
|---------------------|--|-----------------|---------------|
| verboseLogging      | Level of log verbosity. 1 - Errors, 2 - Warnings, 3 - Info, 4 - Debug. | "Value::Int1"   | 0             |
| betaMode            | Enable experimental features (0 - Off, 1 - On).                        | "Value::Int1"   | 0             |
| defaultDetectorType | Detector type: FaceDetV1, FaceDetV2, FaceDetV3.                        | "Value::String" | 0             |

Verbosity level sets the upper limit of what type of messages may be printed out by the Luna SDK. For example, if user set verboseLogging to 3, it means that Errors, Warnings and Info messages will be printed out to the console. Verbose level of 0 indicates that there are no logging messages printed out at all.

#### Example:

```
<section name="system">
  <param name="verboseLogging" type="Value::Int1" x="0" />
  <param name="betaMode" type="Value::Int1" x="0" />
  <param name="detectorType" type="Value::String" text="FaceDetV1" />
</section>
```

## Descriptor factory settings

Descriptor factory is a facility that creates descriptor extractors and matchers. Both of them utilize algorithms that require a number of coefficients (“weights”) to operate properly.

| Parameter                   | Description  | Type            | Default value |
|-----------------------------|--|-----------------|---------------|
| model                       | CNN face descriptor version.<br>Possible values: 54, 56, 57, 58, 59  | "Value::Int1"   | 54            |
| useMobileNet                | MobileNet is faster but less accurate. Possible values: 0 - don't use mobile net version, 1 - use mobile net version.                                    | "Value::Int1"   | 0             |
| distance                    | Distance between descriptors on matching. L1 faster, L2 make better precision. Possible values: L1, L2.  | "Value::String" | "L2"          |
| descriptorCountWarningLevel | Threshold, that limits the ratio of created descriptors to the amount, defined by your license. When the threshold is exceeded, FSDK prints the warning. | "Value::Float1" | 0.9           |
| calcSimilarity              | Enable similarity calculation during matching process. Possible values: 1 - enable, 0 - disable.   | "Value::Int1"   | 1             |
| calcDistanceSqrt            | Enable calculation of the square root of distance. Possible values: 1 - enable, 0 - disable  | "Value::Int1"   | 1             |

Models with versions 54, 56 and 57 support just L2 distance.

### Example:

```
<section name="DescriptorFactory::Settings">
```

```
<param name="model" type="Value::Int1" x="54" />
<param name="useMobileNet" type="Value::Int1" x="0" />
<param name="distance" type="Value::String" text="L2" />
<param name="descriptorCountWarningLevel" type="Value::Float1" x="0.9"
/>
<param name="calcSimilarity" type="Value::Int1" x="1" />
</section>
```

## FaceDetV3 detector settings

| Parameter                 | Description  | Type            | Default value |
|---------------------------|--|-----------------|---------------|
| ScoreThreshold            | Detection score threshold (GRB) in [0..1] range.         | "Value::Float1" | 0.904         |
| ScoreThresholdIR          | Detection score threshold (InfraRed) in [0..1] range.    | "Value::Float1" | 0.784         |
| RedetectScoreThreshold    | Redetect score threshold in [0..1] range                 | "Value::Float1" | 0.357         |
| NMSThreshold              | Overlap threshold for NMS in [0..1] range                | "Value::Float1" | 0.3           |
| minFaceSize               | Minimum face size in pixels.                             | "Value::Int1"   | 50            |
| nms                       | Type of NMS: mean or best                                | "Value::String" | mean          |
| RedetectTensorSize        | Target face after preprocessing for redetect             | "Value::Int1"   | 80            |
|                           | Non-public parameter. Do not change.                     |                 |               |
| RedetectFaceTargetSize    | Target face size for redetect                            | "Value::Int1"   | 45            |
|                           | Non-public parameter. Do not change.                     |                 |               |
| padding                   | Extension of rectangle for RGB mode. Do not change.      | "Value::Float4" | see below     |
| paddingIR                 | Extension of rectangle for InfraRed mode. Do not change. | "Value::Float4" | see below     |
| planPrefix                | Plan prefix  | "Value::String" | FaceDet_v3_5  |
| cropPaddingAlignment      | Non-public parameter. Do not change.                     | "Value::Int1"   | 64            |
| batchCapacity             | Non-public parameter. Do not change.                     | "Value::Int1"   | 16            |
| concurrentBatchSubmission | Non-public parameter. Do not change.                     | "Value::Int1"   | 1             |



| Parameter     | Description                          | Type            | Default value |
|---------------|--------------------------------------|-----------------|---------------|
| detectMean    | Non-public parameter. Do not change. | "Value::Float3" | see below     |
| detectSigma   | Non-public parameter. Do not change. | "Value::Float3" | see below     |
| redetectMean  | Non-public parameter. Do not change. | "Value::Float3" | see below     |
| redetectSigma | Non-public parameter. Do not change. | "Value::Float3" | see below     |

```

<section name="FaceDetV3::Settings">
  <param name="ScoreThreshold" type="Value::Float1" x="0.89"/> <!--
    used for RGB mode -->
  <param name="ScoreThresholdIR" type="Value::Float1" x="0.784"/> <!--
    used for InfraRed mode -->
  <param name="RedetectScoreThreshold" type="Value::Float1" x="0.357"/>
  <param name="NMSThreshold" type="Value::Float1" x="0.3"/>
  <param name="minFaceSize" type="Value::Int1" x="50" />
  <param name="nms" type="Value::String" text="mean"/> <!-- best, mean
    -->
  <param name="RedetectTensorSize" type="Value::Int1" x="80"/>
  <param name="RedetectFaceTargetSize" type="Value::Int1" x="45"/>
  <param name="paddings" type="Value::Float4" x="-0.18685804" y="
    0.09821641" z="0.199056897" w="0.07416578" />
  <param name="paddingsIR" type="Value::Float4" x="-0.12208561" y="
    0.12426723" z="0.15508278" w="0.06038743" />
  <param name="planPrefix" type="Value::String" text="FaceDet_v3_7" />
  <param name="cropPaddingAlignment" type="Value::Int1" x="64" />
  <param name="batchCapacity" type="Value::Int1" x="16" />
  <param name="concurrentBatchSubmission" type="Value::Int1" x="1" />
  <param name="detectMean" type="Value::Float3" x="0.0" y="0.0" z="0.0"
    />
  <param name="detectSigma" type="Value::Float3" x="0.0" y="0.0" z="0.0"
    />
  <param name="redetectMean" type="Value::Float3" x="0.0" y="0.0" z="0.0"
    />
  <param name="redetectSigma" type="Value::Float3" x="0.0" y="0.0" z="0.0"
    />
</section>

```

## FaceDetV1 detector settings

| Parameter         | Description                            | Type            | Default value |
|-------------------|--|-----------------|---------------|
| FirstThreshold    | 1-st threshold in [0..1] range.        | "Value::Float1" | 0.6           |
| SecondThreshold   | 2-nd threshold in [0..1] range.        | "Value::Float1" | 0.7           |
| ThirdThreshold    | 3-d threshold in [0..1] range.         | "Value::Float1" | 0.6           |
| minFaceSize       | Minimum face size in pixels.           | "Value::Int1"   | 50            |
| scaleFactor       | Image scale factor.                    | "Value::Float1" | 0.7           |
| paddings          | Extension of rectangle. Do not change. | "Value::Float4" | see below     |
| redetectTolerance | Redetection threshold                  | "Value::Int1"   | 0             |
| useLNet           | Whether to use LNet or not.            | "Value::Int"    | 1             |

MinSize and scaleFactor accelerate face detection at the cost of lower recall for smaller faces

### Example:

```
<section name="FaceDetV1::Settings">
  <param name="FirstThreshold" type="Value::Float1" x="0.6"/>
  <param name="SecondThreshold" type="Value::Float1" x="0.7"/>
  <param name="ThirdThreshold" type="Value::Float1" x="0.93"/>
  <param name="minFaceSize" type="Value::Int1" x="50" />
  <param name="scaleFactor" type="Value::Float1" x="0.7" />
  <param name="paddings" type="Value::Float4" x="-0.20099958" y="
    0.10210337" z="0.20363552" w="0.08490226"/>
  <param name="redetectTolerance" type="Value::Int1" x="0" />
  <param name="useLNet" type="Value::Int1" x="1" />
</section>
```

## FaceDetV2 detector settings

| Parameter         | Description                            | Type            | Default value |
|-------------------|--|-----------------|---------------|
| FirstThreshold    | 1-st threshold in [0..1] range.        | "Value::Float1" | 0.6           |
| SecondThreshold   | 2-nd threshold in [0..1] range.        | "Value::Float1" | 0.7           |
| ThirdThreshold    | 3-d threshold in [0..1] range.         | "Value::Float1" | 0.6           |
| minFaceSize       | Minimum face size in pixels.           | "Value::Int1"   | 50            |
| scaleFactor       | Image scale factor.                    | "Value::Float1" | 0.7           |
| paddings          | Extension of rectangle. Do not change. | "Value::Float4" | see below     |
| redetectTolerance | Redetection threshold                  | "Value::Int1"   | 0             |
| useLNet           | Whether to use LNet or not.            | "Value::Int"    | 1             |

MinSize and scaleFactor accelerate face detection at the cost of lower recall for smaller faces

### Example:

```
<section name="FaceDetV2::Settings">
  <param name="FirstThreshold" type="Value::Float1" x="0.51385"/>
  <param name="SecondThreshold" type="Value::Float1" x="0.248"/>
  <param name="ThirdThreshold" type="Value::Float1" x="0.76"/>
  <param name="minFaceSize" type="Value::Int1" x="50" />
  <param name="scaleFactor" type="Value::Float1" x="0.7" />
  <param name="paddings" type="Value::Float4" x="-0.20099958" y="
    0.10210337" z="0.20363552" w="0.08490226" />
  <param name="redetectTolerance" type="Value::Int1" x="0" />
  <param name="useLNet" type="Value::Int1" x="0" />
</section>
```

### **LNet**

This group of parameters is non-public. Do not change any of the parameters.

### **LNetIR**

This group of parameters is non-public. Do not change any of the parameters.

### **SLNet**

This group of parameters is non-public. Do not change any of the parameters.

## HumanDetector settings

Human body detector.

| Parameter                 | Type            | Default value |
|---------------------------|-----------------|---------------|
| ScoreThreshold            | "Value::Float1" | x="0.4"       |
| RedetectScoreThreshold    | "Value::Float1" | x="0.12"      |
| NMSThreshold              | "Value::Float1" | x="0.4"       |
| RedetectNMSThreshold      | "Value::Float1" | x="0.3"       |
| imageSize                 | "Value::Int1"   | x="640"       |
| nms                       | "Value::String" | text="mean"   |
| RedetectNMS               | "Value::String" | text="mean"   |
| humanLandmarks17Threshold | "Value::Float1" | x="0.2"       |

### Example:

```
<section name="HumanDetector::Settings">
  <param name="ScoreThreshold" type="Value::Float1" x="0.4"/>
  <param name="RedetectScoreThreshold" type="Value::Float1" x="0.12"/>
  <param name="NMSThreshold" type="Value::Float1" x="0.4"/>
  <param name="RedetectNMSThreshold" type="Value::Float1" x="0.3"/>
  <param name="imageSize" type="Value::Int1" x="640"/>
  <param name="nms" type="Value::String" text="mean"/> <!-- best, mean -->
  <param name="RedetectNMS" type="Value::String" text="mean"/> <!-- best,
    mean -->
  <param name="humanLandmarks17Threshold" type="Value::Float1" x="0.2"/>
</section>
```

## IndexBuilder settings

HNSW index can be built with descriptors batches and used to search nearest descriptor neighbors very fast.

| Parameter    | Description  | Type          | Default value |
|--------------|--|---------------|---------------|
| numThreads   | Number of threads to use on build. If 0 or less, use std::hardware_concurrency value.  | "Value::Int1" | 0             |
| construction | Internal construction value. The greater it is, the better is graph, but slower construction. DO NOT CHANGE, unless you know what you are doing. | "Value::Int1" | 1600          |
| search       | Internal search value. Greater value means slower but more complete search. DO NOT CHANGE, unless you know what you are doing.                   | "Value::Int1" | 1000          |

### Example:

```
<section name="IndexBuilder::Settings">  
  <param name="numThreads" type="Value::Int1" x="0" />  
  <param name="construction" type="Value::Int1" x="2000" />  
  <param name="search" type="Value::Int1" x="6000" />  
</section>
```

## Quality estimator settings

Quality estimator looks at several image parameters, like lightness (think overexposure), darkness (think underexposure), blurriness, illumination uniformity value, specularity value. Every float value is comparing with according threshold.

| Parameter             | Type            | Default value |
|-----------------------|-----------------|---------------|
| blurThreshold         | "Value::Float1" | x="0.61"      |
| lightThreshold        | "Value::Float1" | x="0.57"      |
| darknessThreshold     | "Value::Float1" | x="0.50"      |
| illuminationThreshold | "Value::Float1" | x="0.1"       |
| specularityThreshold  | "Value::Float1" | x="0.1"       |

### Example:

```
<section name="QualityEstimator::Settings">
  <param name="blurThreshold" type="Value::Float1" x="0.61"/>
  <param name="lightThreshold" type="Value::Float1" x="0.57"/>
  <param name="darknessThreshold" type="Value::Float1" x="0.50"/>
  <param name="illuminationThreshold" type="Value::Float1" x="0.1"/>
  <param name="specularityThreshold" type="Value::Float1" x="0.1"/>
</section>
```

## HeadPoseEstimator settings

HeadPose estimator is able to compute head pose angles in two different ways.

The first one estimates angles by 68-point face-alignment results.

The second one uses raw input image data.

Configuration block listed below allows user to define which method to use. Default configuration settings enables both estimation methods.

| Parameter                | Type          | Default value |
|--------------------------|---------------|---------------|
| useEstimationByImage     | "Value::Int1" | 1             |
| useEstimationByLandmarks | "Value::Int1" | 1             |

### Example:

```
<section name="HeadPoseEstimator::Settings">  
  <param name="useEstimationByImage" type="Value::Int1" x="1"/>  
  <param name="useEstimationByLandmarks" type="Value::Int1" x="0"/>  
</section>
```



## EyeEstimator settings

This estimator aims to determine:

- Eye state: Open, Closed, Occluded;
- Precise eye iris location as an array of landmarks;
- Precise eyelid location as an array of landmarks.

To determine more exact eye state additional auxiliary model `eye_status_estimation_flwr*.plan` is used. You can enable this auxiliary model through config (`faceengine.conf`).

| Parameter                  | Description     | Type                       | Default value |
|----------------------------|-----------------|----------------------------|---------------|
| <code>useStatusPlan</code> | 0 - Off, 1 - On | <code>"Value::Int1"</code> | 1             |

### Example:

```
<section name="EyeEstimator::Settings">
  <param name="useStatusPlan" type="Value::Int1" x="1"/>
</section>
```

## AttributeEstimator settings

This estimator is able to estimate many person attributes such as:

- person's age;
- gender: male, female;

Some of estimator result values depends on threshold values listed below.

| Parameter       | Description                       | Type                 | Default value |
|-----------------|-----------------------------------|----------------------|---------------|
| genderThreshold | gender threshold in [0..1] range. | "Value::Float1"<br>" | 0.5           |
| adultThreshold  | adult threshold in [0..1] range.  | "Value::Float1"<br>" | 0.2           |

### Example:

```
<section name="AttributeEstimator::Settings">  
  <param name="genderThreshold" type="Value::Float1" x="0.5"/>  
  <param name="adultThreshold" type="Value::Float1" x="0.2"/>  
</section>
```

## GlassesEstimator settings

Glasses estimator estimates what types of glasses, if any, person is currently wearing. Quality of estimation depends on threshold values listed below. These threshold values set to optimal by default.

| Parameter           | Description                           | Type            | Default value |
|---------------------|---------------------------------------|-----------------|---------------|
| noGlassesThreshold  | noGlasses threshold in [0..1] range.  | "Value::Float1" | 0.986         |
| eyeGlassesThreshold | eyeGlasses threshold in [0..1] range. | "Value::Float1" | 0.57          |
| sunGlassesThreshold | sunGlasses threshold in [0..1] range. | "Value::Float1" | 0.506         |

### Example:

```
<section name="GlassesEstimator::Settings">
  <param name="noGlassesThreshold" type="Value::Float1" x="0.986"/>
  <param name="eyeGlassesThreshold" type="Value::Float1" x="0.57"/>
  <param name="sunGlassesThreshold" type="Value::Float1" x="0.506"/>
</section>
```

## OverlapEstimator settings

This estimator tells whether the face is overlapped by any object.

It returns a structure with 2 fields. The first is the value of overlapping in the range from 0.0 (is not overlapped) to 1.0 (maximum, overlapped), the second is a boolean answer.

The boolean answer depends on the threshold listed below. If the value is greater than the threshold, the answer returns true, else false.

| Parameter        | Description                        | Type                 | Default value |
|------------------|------------------------------------|----------------------|---------------|
| overlapThreshold | overlap threshold in [0..1] range. | "Value::Float1"<br>" | 0.01          |

### Example:

```
<section name="OverlapEstimator::Settings">  
  <param name="overlapThreshold" type="Value::Float1" x="0.01"/>  
</section>
```

## ChildEstimator settings

This estimator tells whether the person is child or not.

Child is a person who is younger than 18 years old.

The estimator returns a structure with 2 fields. The first is the score in the range from 0.0 (is an adult) to 1.0 (maximum, is a child), the second is a boolean answer.

The boolean answer depends on the threshold listed below. If the value is less than the threshold, then true is returned (the person is a child), else false (the person is an adult).

| Parameter      | Description                | Type            | Default value |
|----------------|----------------------------|-----------------|---------------|
| childThreshold | threshold in [0..1] range. | "Value::Float1" | 0.8508        |

### Example:

```
<section name="ChildEstimator::Settings">  
  <param name="ChildThreshold" type="Value::Float1" x="0.8508"/>  
</section>
```

## LivenessIREstimator settings

This estimator determines whether the person's face is real or fake (photo, printed image).

Image must be received from infra-red camera.

The estimator returns a boolean answer (true - is real, false - is fake).

Estimator can be used in "universal" and "ambarella" modes. The mode is chosen depending on the camera type. Thresholds are listed below.

| Parameter            | Description                | Type            | Default value |
|----------------------|----------------------------|-----------------|---------------|
| name                 | universal                  | "Value::String" | universal     |
| irUniversalThreshold | threshold in [0..1] range. | "Value::Float1" | 0.5328        |
| irAmbarellaThreshold | threshold in [0..1] range. | "Value::Float1" | 0.76          |

```
<section name="LivenessIREstimator::Settings">
  <param name="name" type="Value::String" x="universal"/>
  <param name="irUniversalThreshold" type="Value::Float1" x="0.5328"/>
  <param name="irAmbarellaThreshold" type="Value::Float1" x="0.76"/>
</section>
```

## HeadAndShouldersLivenessEstimator settings

This estimator tells whether the person's face is real or fake (photo, printed image). Thresholds are listed below.

| Parameter            | Description                    | Type                 | Default value |
|----------------------|--------------------------------|----------------------|---------------|
| headWidthKoeff       | threshold in [0.5..2.0] range  | "Value::Float1"<br>" | 1.0           |
| headHeightKoeff      | threshold in [0.5..2.0] range. | "Value::Float1"<br>" | 1.0           |
| shouldersWidthKoeff  | threshold in [0.5..2.0] range. | "Value::Float1"<br>" | 0.75          |
| shouldersHeightKoeff | threshold in [1.5..5.0] range. | "Value::Float1"<br>" | 3.0           |

```
<section name="HeadAndShouldersLivenessEstimator::Settings">  
  <param name="headWidthKoeff" type="Value::Float1" x="1.0"/>  
  <param name="headHeightKoeff" type="Value::Float1" x="1.0"/>  
  <param name="shouldersWidthKoeff" type="Value::Float1" x="0.75"/>  
  <param name="shouldersHeightKoeff" type="Value::Float1" x="3.0"/>  
</section>
```

## Mouth Estimator settings

Mouth estimator predicts predominant mouth state.

Estimator accuracy depends on thresholds listed below.

| Parameter          | Description                | Type            | Default value |
|--------------------|----------------------------|-----------------|---------------|
| occlusionThreshold | threshold in [0..1] range  | "Value::Float1" | 0.3           |
| smileThreshold     | threshold in [0..1] range. | "Value::Float1" | 0.55          |
| openThreshold      | threshold in [0..1] range. | "Value::Float1" | 0.64          |

```
<section name="MouthEstimator::Settings">  
  <param name="occlusionThreshold" type="Value::Float1" x="0.3"/>  
  <param name="smileThreshold" type="Value::Float1" x="0.55"/>  
  <param name="openThreshold" type="Value::Float1" x="0.64"/>  
</section>
```



## Medical mask estimator settings

Medical mask estimator predicts predominant mask features.

Estimator accuracy depends on thresholds listed below.

If accuracy (low FPR) is more important, TPR could be sacrificed by heightening the threshold.

Corresponding FPR and TPR values are also listed in the table below.

**Table 18:** “Thresholds for MedicalMaskEstimation”

| Parameter    | Description  | Type            | Threshold range | FPR range     | TPR range    |
|--------------|--------------|-----------------|-----------------|---------------|--------------|
| mask         | range [0..1] | "Value::Float1" | 0.25 – 0.818    | 0.1 – 0.05    | 0.987 – 0.95 |
| noMask       | range [0..1] | "Value::Float1" | 0.108 – 0.792   | 0.05 – 0.018  | 0.95 – 0.83  |
| occludedFace | range [0..1] | "Value::Float1" | 0.101 – 0.498   | 0.075 – 0.035 | 0.87 – 0.79  |

**Table 19:** “Thresholds for MedicalMaskEstimationExtended”

| Parameter              | Description  | Type            | Threshold range | FPR range     | TPR range    |
|------------------------|--------------|-----------------|-----------------|---------------|--------------|
| maskExtended           | range [0..1] | "Value::Float1" | 0.36 – 0.9      | 0.035 – 0.01  | 0.99 – 0.917 |
| noMaskExtended         | range [0..1] | "Value::Float1" | 0.108 – 0.792   | 0.05 – 0.018  | 0.95 – 0.83  |
| maskNotInPlaceExtended | range [0..1] | "Value::Float1" | 0.18 – 0.76     | 0.064 – 0.02  | 0.94 – 0.77  |
| occludedFaceExtended   | range [0..1] | "Value::Float1" | 0.101 – 0.498   | 0.075 – 0.035 | 0.87 – 0.79  |

### Example:

```
<section name="MedicalMaskEstimator::Settings">  
  <param name="maskExtendedThreshold" type="Value::Float1" x="0.36"/>  
</section>
```

```
<param name="noMaskExtendedThreshold" type="Value::Float1" x="0.108"
/>
<param name="maskNotInPlaceExtendedThreshold" type="Value::Float1" x
="0.18"/>
<param name="occludedFaceExtendedThreshold" type="Value::Float1" x="
0.101"/>
<param name="maskThreshold" type="Value::Float1" x="0.25"/>
<param name="noMaskThreshold" type="Value::Float1" x="0.108"/>
<param name="occludedFaceThreshold" type="Value::Float1" x="0.101"/>
</section>
```

## Depth Estimator settings

Depth estimator performs liveness check via depth image. It exposes different threshold parameters where each one of them let you configure estimator for your specific use case.

| Parameter           | Description  | Type            | Default value |
|---------------------|--|-----------------|---------------|
| maxDepthThreshold   | maximum depth distance threshold in mm. Should be in [0..inf] range.                     | "Value::Float1" | 3000          |
| minDepthThreshold   | minimum depth distance threshold in mm. Should be in [0..maxDepthThreshold] range.       | "Value::Float1" | 100           |
| zeroDepthThreshold  | percentage of zero pixels in input image. Threshold in [0..1] range.                     | "Value::Float1" | 0.66          |
| confidenceThreshold | score threshold above which person is considered to be alive. Threshold in [0..1] range. | "Value::Float1" | 0.89          |

```
<section name="DepthEstimator::Settings">
  <param name="maxDepthThreshold" type="Value::Float1" x="3000"/>
  <param name="minDepthThreshold" type="Value::Float1" x="100"/>
  <param name="zeroDepthThreshold" type="Value::Float1" x="0.66"/>
  <param name="confidenceThreshold" type="Value::Float1" x="0.89"/>
</section>
```

## LivenessFlyingFaces Estimator settings

This estimator tells whether the person's face is real or fake (photo, printed image).

It returns a structure with 2 fields.

The first one is the value in the range from 0.0 (is not real) to 1.0 (maximum, real), the second is a boolean answer.

The boolean answer depends on the "realThreshold". If the value is greater than the threshold, the answer returns true, else false.

| Parameter        | Description                  | Type            | Default value |
|------------------|------------------------------|-----------------|---------------|
| realThreshold    | threshold in [0..1] range.   | "Value::Float1" | 0.98          |
| aggregationCoeff | coefficient in [0..1] range. | "Value::Float1" | 0.5           |

### Example:

```
<section name="LivenessFlyingFacesEstimator::Settings">
  <param name="realThreshold" type="Value::Float1" x="0.98"/>
  <param name="aggregationCoeff" type="Value::Float1" x="0.5"/>
</section>
```

## LivenessRGBM Estimator settings

This estimator tells whether the person's face is real or fake (photo, printed image).

It returns a structure with 2 fields.

The first one is the value in the range from 0.0 (is not real) to 1.0 (maximum, real). The second is a boolean answer.

The boolean answer depends on the "threshold". If the value is greater than the threshold, the answer returns true, else false.

This estimator work is based on background accumulation. So the "backgroundCount" parameter is the amount of the frames for the background calculation.

Other parameters are implementation specific, they are not recommended to change.

| Parameter       | Description                          | Type            | Default value |
|-----------------|--------------------------------------|-----------------|---------------|
| backgroundCount | frames count                         | "Value::Int1"   | 100           |
| threshold       | threshold                            | "Value::Float1" | 0.8           |
| coeff1          | Non-public parameter. Do not change. | "Value::Float1" | "0.222"       |
| coeff2          | Non-public parameter. Do not change. | "Value::Float1" | "0.222"       |

### Example:

```
<section name="LivenessRGBMEstimator::Settings">
  <param name="backgroundCount" type="Value::Int1" x="100"/>
  <param name="threshold" type="Value::Float1" x="0.8"/>
  <param name="coeff1" type="Value::Float1" x="0.222"/>
  <param name="coeff2" type="Value::Float1" x="0.222"/>
</section>
```

## LivenessOneShotRGBEstimator

This estimator tells whether the person's face is real or fake (photo, printed image). Thresholds are listed below.

LivenessOneShotRGBEstimator supports images, which are captured on Mobile devices or Webcam (PC or laptop).

Image resolution minimum requirements:

- Mobile devices - 720 × 960 px
- Webcam (PC or laptop) - 1280 x 720 px

| Parameter        | Description                | Type            | Default value |
|------------------|----------------------------|-----------------|---------------|
| realThreshold    | threshold in [0..1] range. | "Value::Float1" | 0.8           |
| qualityThreshold | Default value is 0.        | "Value::Float1" | 0             |

```
<section name="LivenessOneShotRGBEstimator::Settings">
  <param name="realThreshold" type="Value::Float1" x="0.88"/>
  <param name="qualityThreshold" type="Value::Float1" x="0" />
</section>
```

## Credibility Estimator settings

Credibility estimator is trained to predict reliability of a person. It does so by returning a score value between [0;1] which will be closer to 1 if a person is more likely to be reliable and closer to 0 otherwise. Along with the output score value estimator also returns an enum value, which will give a plain answer if a person is reliable or not for a user convenience. Credibility estimator sets this enum value by comparing an output score with a reliability threshold value listed in faceengine.conf file. User can modify this threshold in CredibilityEstimator::Settings section:

| Parameter         | Description | Type            | Default value |
|-------------------|-------------|-----------------|---------------|
| reliableThreshold | threshold   | "Value::Float1" | 0.5           |

### Example:

```
<section name="CredibilityEstimator::Settings">  
  <param name="reliableThreshold" type="Value::Float1" x="0.5"/>  
</section>
```

## Runtime settings

Runtime configuration file provides parameters that user can tweak to achieve optimal performance of their app.

The name of runtime configuration file is runtime.conf and its placed in data directory. Its settings are described below:

| Parameter         | Description   | Type            | Default value |
|-------------------|---|-----------------|---------------|
| cpuClass          | Class of cpu by supported instructions - cpu, sse4, avx, avx2, arm, auto.   | "Value::String" | "auto"        |
| deviceClass       | Execution device type - cpu, gpu.   | "Value::String" | "cpu"         |
| numThreads        | Number of worker threads. Default: number of CPU logical cores.   | "Value::Int1"   | -1            |
| verboseLogging    | Level of log verbosity. 1 - Errors, 2 - Warnings, 3 - Info, 4 - Debug.  | "Value::Int1"   | 0             |
| numComputeStreams | Number of streams; Increases performance, but works only with new versions of NVIDIA drivers (375.82, 384.59 and more recent). Don't increase it with older version of NVIDIA driver. | "Value::Int1"   | 1             |
| programCacheSize  | Maximum number of Program objects in cache.   | "Value::Int1"   | 128           |
| defaultGpuDevice  | Default GPU device number   | "Value::Int1"   | 0             |

Verbosity level sets the upper limit of what type of messages may be printed out. For example, if user set verboseLogging to 3, it means that Errors, Warnings and Info messages will be printed out to the console. Verbose level of 0 indicates that there are no logging messages printed out at all.

In case of GPU usage the numThreads value should be at least == 2 or -1. If this requirement is violated, further behavior is undefined.



Increasing the `programCacheSize` increases memory usage and potentially improves performance. Be careful, too large a value of this parameter can lead to a crash due to insufficient memory.

**Example:**

```
<section name="Runtime">
  <param name="cpuClass" type="Value::String" text="auto" />
  <param name="deviceClass" type="Value::String" text="cpu" />
  <param name="numThreads" type="Value::Int1" x="-1" />
  <param name="verboseLogging" type="Value::Int1" x="0" />
  <param name="numComputeStreams" type="Value::Int1" x="4" />
  <param name="programCacheSize" type="Value::Int1" x="128" />
  <param name="defaultGpuDevice" type="Value::Int1" x="0" />
</section>
```

*Note:* Setting `<param name="numThreads" type="Value::Int1" x="-1"/>` means that will be taken the maximum number of available threads. This number of threads is equal to according number of available processor cores.

*Note:* Setting `<param name="defaultGpuDevice" type="Value::Int1" x="-1"/>` means disable GPU runtime initialisation. Set it only with `deviceClass == cpu`.