

Configuration Guide

Contents

Configuration Guide	3
Configuration file location	3
Settings	5
System settings	5
Descriptor factory settings	6
FaceDetV3 detector settings	7
FaceDetV1 detector settings	9
FaceDetV2 detector settings	10
LNet	11
LNetIR	11
SLNet	11
HumanDetector settings	11
IndexBuilder settings	13
Quality estimator settings	13
HeadPoseEstimator settings	15
BestShotQualityEstimator 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	26
LivenessFlyingFaces Estimator settings	27
LivenessRGBM Estimator settings	28
Credibility Estimator settings	29
Licensing	30
Licensing settings	30

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.

NOTE: 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.

NOTE: 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"	2
betaMode	Enable experimental features (0 - Off, 1 - On).	"Value::Int1"	0
defaultDetectorType	Detector type: FaceDetV1, FaceDetV2, FaceDetV3.	"Value::String"	0

Note: 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="2" />
  <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. Possible values: 46, 52, 54, 56, 57, 58	"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::Float1"	"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

Note: Models with versions 46, 52, 54, 56 and 57 support just L2 distance.

Note: Version 46 and 52 are deprecated since LUNA SDK release v.4.1.0. Use 54, 56 or 57 versions in new projects.

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" />
</section>
```

FaceDetV3 detector settings

Parameter	Description	Type	Default value
ScoreThreshold	Detection score threshold (RGB) 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
maxFaceSize	Maximum face size in pixels. (Must not be greater then minFaceSize*32)	"Value::Int1"	640
nms	Type of NMS: mean or best	"Value::String"	mean
RedetectTensorSize	Target face after preprocessing for redetect	"Value::Int1"	80
RedetectFaceTargetSize	Target face size for redetect	"Value::Int1"	45
paddings	Extension of rectangle for RGB mode. Do not change.	"Value::Float4"	see below
paddingsIR	Extension of rectangle for InfraRed mode. Do not change.	"Value::Float4"	see below
planPrefix	Plan prefix	"Value::String"	FaceDet_v3_5
useOrientationMode	Use mode for rotated origin images or not*	"Value::Int1"	x="0"
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

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

*) When the "useOrientationMode" value is equal to "1", the mode for rotated origin images is enabled.

If the image has standard orientation, the image is processed as if the value is equal to "0".

If the image is rotated to 90, 180, or 270 degrees, the detector defines that. Then the image is rotated to the standard orientation and the detection is performed on the rotated image.

The rotated image is saved. The detection and points will be in the coordinates of the rotated image. The source image is not changed.

```
<section name="FaceDetV3::Settings">
  <param name="ScoreThreshold" type="Value::Float1" x="0.904"/>
  <param name="ScoreThresholdIR" type="Value::Float1" x="0.784"/>
  <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="20" />
  <param name="maxFaceSize" type="Value::Int1" x="640" />
  <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_5" />
  <param name="useOrientationMode" type="Value::Int1" x="0" />
  <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="paddings" type="Value::Float4" x="-0.20099958" y="
    0.10210337" z="0.20363552" w="0.08490226" />
  <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="1" />
</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="0.25"

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="0.25"/>
```

</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"	2000
search	Internal search value. Greater value means slower but more complete search. DO NOT CHANGE, unless you know what you are doing.	"Value::Int1"	6000

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"

Parameter	Type	Default value
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>
```

BestShotQualityEstimator settings

This estimator includes HeadPose and AGS estimators.

The “runSubestimatorsConcurrently” parameter determines whether the two estimators are working consequentially or in parallel upon calling.

Parameter	Type	Default value
runSubestimatorsConcurrently	"Value::Int1"	0

Example:

```
<section name="BestShotQualityEstimator::Settings">  
  <param name="runSubestimatorsConcurrently" 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;
- person's ethnicity.

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" "	0.5
adultThreshold	adult threshold in [0..1] range.	"Value::Float1" "	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" "	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", "ambarella", and "verme" modes. The mode is chosen depending on the camera type and configured in the "faceengine.conf" file.

Thresholds are listed below.

Parameter	Description	Type	Default value
name	universal, ambarella, verme	"Value::String"	universal
irUniversalThreshold	threshold in [0..1] range.	"Value::Float1"	0.5328
irAmbarellaThreshold	threshold in [0..1] range.	"Value::Float1"	0.76
irVermeThreshold	threshold in [0..1] range.	"Value::Float1"	0.35

```
<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"/>
  <param name="irVermeThreshold" type="Value::Float1" x="0.35"/>
</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" "	1.0
headHeightKoeff	threshold in [0.5..2.0] range.	"Value::Float1" "	1.0
shouldersWidthKoeff	threshold in [0.5..2.0] range.	"Value::Float1" "	0.75
shouldersHeightKoeff	threshold in [1.5..5.0] range.	"Value::Float1" "	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.

Parameter	Description	Type	Default value
maskThreshold	range [0..1]	"Value::Float1"	0.31532
noMaskThreshold	range [0..1]	"Value::Float1"	0.43823
maskNotInPlace	range [0..1]	"Value::Float1"	0.14172
occludedFaceThreshold	range [0..1]	"Value::Float1"	0.05439

```
<section name="MedicalMaskEstimator::Settings">
  <param name="maskThreshold" type="Value::Float1" x="0.31532"/>
  <param name="noMaskThreshold" type="Value::Float1" x="0.43823"/>
  <param name="maskNotInPlace" type="Value::Float1" x="0.14172"/>
  <param name="occludedFaceThreshold" type="Value::Float1" x="0.05439"
/>
</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"	0.945

Example:

```
<section name="LivenessFlyingFacesEstimator::Settings">  
  <param name="realThreshold" type="Value::Float1" x="0.945"/>  
</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
threshold	threshold	"Value::Float1"	0.8
backgroundCount	frames count	"Value::Int1"	100
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>
```

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>
```

Licensing

Licensing settings

Licensing configuration options are specified via license.conf file which is basically an XML document with special tag formatting. This file is mandatory for license activation. Please, fill it with correct values before FaceEngine usage.

NOTE! 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.

The location where the license config file is found is similar with faceengine.conf location.

License activation and next processing requires parameters listed below.

Parameter	Description	Type	Default value
Server	Activation server URL	"Value::String"	(empty)
EID	Entitlement ID	"Value::String"	(empty)
ProductID	Product ID	"Value::String"	(empty)
Filename	Default license filename	"Value::String"	(empty)
ContainerMode	activate license in containers	"Value::Int1"	0

A little bit more description: Server, EID and ProductID - please, request this information and write it in the config. It is mandatory for activation procedure.

Filename - name of the file to save license after activation. At the next activations on the same device license will be read from this file.

ContainerMode - flag that allows using licensed Luna SDK in docker containers.

```
<section name="Licensing::Settings">
  <param name="Server" type="Value::String" text=""/>
  <param name="EID" type="Value::String" text=""/>
  <param name="ProductID" type="Value::String" text=""/>
  <param name="Filename" type="Value::String" text=""/>
  <param name="ContainerMode" type="Value::Int1" x="0"/>
</section>
```