Configuration Guide

Contents

Runtime settings **56**

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

> 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: 58, 59, 60, 62	`"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"`
descriptorCount WarningLevel	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 58, 59, 60 support just L2 distance.

Example:

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

```xml
        <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" />
        <param name="calcDistanceSqrt" type="Value::Int1" x="1" />
    </section>
```

FaceDetV3 detector settings

Parameter	Description	Type	Default value
ScoreThreshold	Detection score threshold (RGB) in [0..1] range.	`"Value ::Float1"`	0.5
ScoreThresholdNPU	Detection score threshold (RGB) in [0..1] range.	`"Value ::Float1"`	0.89
ScoreThresholdIR	Detection score threshold (InfraRed) in [0..1] range.	`"Value ::Float1"`	0.38
RedetectScoreThreshold	Redetect score threshold in [0..1] range	`"Value ::Float1"`	0.3
NMSThreshold	Overlap threshold for NMS in [0..1] range	`"Value ::Float1"`	0.35
NMSThresholdNPU	Overlap threshold for NMS in [0..1] range	`"Value ::Float1"`	0.35
intraNMSThreshold	Overlap threshold for intra NMS in [0..1] range	`"Value ::Float1"`	0.35
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"`	64

Parameter	Description	Type	Default value
	Non-public parameter. Do not change.		
paddings	Extension of rectangle for RGB mode. Do not change.	"Value :: Float4"	see below
planPrefix	Plan prefix	"Value :: String"	FaceDet_v3_a5
pyramidAlgorithm	Turn On/Off the pyramid based algorithm	"Value ::Int1"	1
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
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

```xml
<section name="FaceDetV3::Settings">
    <param name="ScoreThreshold" type="Value::Float1" x="0.5"/>   <!-- used
        for RGB mode -->
    <param name="ScoreThresholdNPU" type="Value::Float1" x="0.89"/>   <!--
        used for RGB mode -->
    <param name="ScoreThresholdIR" type="Value::Float1" x="0.38"/> <!-- used
        for InfraRed mode -->
```

```xml
        <param name="RedetectScoreThreshold" type="Value::Float1" x="0.3"/>
        <param name="NMSThreshold" type="Value::Float1" x="0.35"/>
        <param name="NMSThresholdNPU" type="Value::Float1" x="0.35"/>
        <param name="intraNMSThreshold" type="Value::Float1" x="0.35"/>
        <param name="minFaceSize" type="Value::Int1" x="50" />
        <param name="pyramidAlgorithm" type="Value::Int1" x="1" />
        <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="64"/>
        <param name="paddings" type="Value::Float4" x="-0.18685804" y="
           0.09821641" z="0.199056897" w="0.07416578" />
        <param name="planPrefix" type="Value::String" text="FaceDet_v3_a5" />
        <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>
```

FaceDetV3m detector settings

Parameter	Description	Type	Default value
ScoreThreshold	Detection score threshold (RGB) in [0..1] range.	`"Value :: Float1"`	0.5
ScoreThresholdNPU	Detection score threshold (RGB) in [0..1] range.	`"Value :: Float1"`	0.89
ScoreThresholdIR	Detection score threshold (InfraRed) in [0..1] range.	`"Value :: Float1"`	0.38
RedetectScoreThreshold	Redetect score threshold in [0..1] range	`"Value :: Float1"`	0.3
NMSThreshold	Overlap threshold for NMS in [0..1] range	`"Value :: Float1"`	0.35
NMSThresholdNPU	Overlap threshold for NMS in [0..1] range	`"Value :: Float1"`	0.35
intraNMSThreshold	Overlap threshold for intra NMS in [0..1] range	`"Value :: Float1"`	0.35
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"`	64

Parameter	Description	Type	Default value
	Non-public parameter. Do not change.		
paddings	Extension of rectangle for RGB mode. Do not change.	`"Value :: Float4"`	see below
planPrefix	Plan prefix	`"Value :: String"`	`FaceDet_v3m`
pyramidAlgorithm	Turn On/Off the pyramid based algorithm	`"Value ::Int1"`	1
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
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="FaceDetV3m::Settings">
        <param name="ScoreThreshold" type="Value::Float1" x="0.5"/>   <!--
           used for RGB mode -->
        <param name="ScoreThresholdNPU" type="Value::Float1" x="0.89"/>
           <!-- used for RGB mode -->
        <param name="ScoreThresholdIR" type="Value::Float1" x="0.38"/> <!--
           used for InfraRed mode -->
```

```xml
        <param name="RedetectScoreThreshold" type="Value::Float1" x="0.3"/>
        <param name="NMSThreshold" type="Value::Float1" x="0.35"/>
        <param name="NMSThresholdNPU" type="Value::Float1" x="0.35"/>
        <param name="intraNMSThreshold" type="Value::Float1" x="0.35"/>
        <param name="minFaceSize" type="Value::Int1" x="50" />
        <param name="pyramidAlgorithm" type="Value::Int1" x="1" />
        <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="64"/>
        <param name="paddings" type="Value::Float4" x="-0.14193088" y="
            0.02616714" z="0.15237510" w="0.07140099"/>
        <param name="planPrefix" type="Value::String" text="FaceDet_v3m" />
        <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"`	`0.93`
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:

```xml
<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"`	`0.51385`
SecondThreshold	2-nd threshold in [0..1] range.	`"Value::Float1"`	`0.248`
ThirdThreshold	3-d threshold in [0..1] range.	`"Value::Float1"`	`0.76`
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.

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>
```

HumanDetector settings

Human body detector.

Parameter	Type	Default value
ScoreThreshold	"Value::Float1"	x="0.45"
RedetectScoreThreshold	"Value::Float1"	x="0.12"
NMSThreshold	"Value::Float1"	x="0.4"
RedetectNMSThreshold	"Value::Float1"	x="0.4"
imageSize	"Value::Int1"	x="640"
nms	"Value::String"	text="best"
RedetectNMS	"Value::String"	text="mean"
humanLandmarks17Threshold	"Value::Float1"	x="0.2"
RedetectTensorSize	"Value::Int1"	x="110"
RedetectHumanTargetSize	"Value::Int1"	x="85"

Example:

```xml
<section name="HumanDetector::Settings">
   <param name="ScoreThreshold" type="Value::Float1" x="0.45"/>
   <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="0.4"/>
   <param name="imageSize" type="Value::Int1" x="640"/>
   <param name="nms" type="Value::String" text="best"/> <!-- best, mean -->
   <param name="RedetectNMS" type="Value::String" text="mean"/> <!-- best,
      mean -->
   <param name="humanLandmarks17Threshold" type="Value::Float1" x="0.2"/>
   <param name="RedetectTensorSize" type="Value::Int1" x="110"/>
   <param name="RedetectHumanTargetSize" type="Value::Int1" x="85"/>
</section>
```

Head detector settings

Parameter	Description	Type	Default value
ScoreThreshold	Detection score threshold (RGB) in [0..1] range.	"Value:: Float1"	0.5
NMSThreshold	Overlap threshold for NMS in [0..1] range	"Value:: Float1"	0.35
minHeadSize	Minimum face size in pixels.	"Value:: Int1"	60
nms	Type of NMS: mean or best	"Value:: String"	mean
cropPaddingAlignment	Non-public parameter. Do not change.	"Value:: Int1"	64
batchCapacity	Non-public parameter. Do not change.	"Value:: Int1"	16
concurrentBatchSubmis	Non-public parameter. Do not change.	"Value:: Int1"	1

```xml
<section name="HeadDetector::Settings">
        <param name="ScoreThreshold" type="Value::Float1" x="0.5"/>
        <param name="NMSThreshold" type="Value::Float1" x="0.35"/>
        <param name="minHeadSize" type="Value::Int1" x="60" />
        <param name="nms" type="Value::String" text="mean"/> <!-- best, mean
            -->
        <param name="cropPaddingAlignment" type="Value::Int1" x="64" />
        <param name="batchCapacity" type="Value::Int1" x="16" />
        <param name="concurrentBatchSubmission" type="Value::Int1" x="1" />
</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:

```xml
<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>
```

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"	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>
```

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
useStatusPlan	0 - Off, 1 - On	"Value::Int1"	1

Example:

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

LivenessFPREstimator settings

Thresholds are listed below.

Parameter	Description	Type	Default value
realThreshold	threshold in [0..1] range.	"Value::Float1"	0.6

```
<section name="LivenessFPREstimator::Settings">
    <param name="realThreshold" type="Value::Float1" x="0.6"/>
</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>
```

NIRLivenessEstimator settings

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

Thr image must be received from the infra-red camera.

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

The estimator can be used in two modes. The first mode is fast, while the other is slower but more accurate. By default, recommended to use the more accurate mode, which is 2.

Parameter	Description	Type	Default value
realThreshold	threshold in [0..1] range.	"Value:: Float1"	0.5
defaultEstimatorMode	Configuration of `plan files` usage.	Value::Int1	2

```
<section name="NIRLivenessEstimator::Settings">
        <param name="realThreshold" type="Value::Float1" x="0.5"/>
        <!-- Currently, available values to select the estimation mode are:
            1 and 2. -->
        <param name="defaultEstimatorMode" type="Value::Int1" x="2"/>
</section>
```

NIRLiveness estimation mode.

Currently, the available values to select the estimation mode are: Default, M1 and M2. The scenario `Default` means the mode is specified in config file. @see ISettingsProvider.

Implementation description:

NIRLiveness Estimation Mode Enum

```
enum class NIRLivenessMode {
    Default,  // Specified in config file.
    M1,       // M1.
    M2        // M2.
};
```

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

```xml
<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.

FPR and TPR values are specified for 0.5 threshold

Table 21: "Thresholds for `MouthEstimation`"

Parameter	Description	Type	Default value	Threshold range	TPR	FPR
occlusionThresh	threshold in [0..1] range	"Value::Float1"	0.5	0.4 – 0.6	0.96	0.009
smileThreshold	threshold in [0..1] range.	"Value::Float1"	0.5	0.4 – 0.6	0.97	0.04
openThreshold	threshold in [0..1] range.	"Value::Float1"	0.5	0.4 – 0.6	0.986	0.01

Example:

```xml
<section name="MouthEstimator::Settings">
    <param name="occlusionThreshold" type="Value::Float1" x="0.5"/>
    <param name="smileThreshold" type="Value::Float1" x="0.5"/>
    <param name="openThreshold" type="Value::Float1" x="0.5"/>
</section>
```

DeepFake Estimator settings

This estimator is designed to predict is the detected face on the input image synthetic or not.

Estimator accuracy depends on `realThreshold` and `defaultEstimatorType` listed below.

Table 22: "`DeepFakeEstimator`"

Parameter	Description	Type	Default value
realThreshold	Threshold in [0..1] range.	"Value::Float1"	0.5
defaultEstimatorType	Configuration of `plan files` usage.	Value::Int1	2

Example:

```
<section name="DeepFakeEstimator::Settings">
        <param name="realThreshold" type="Value::Float1" x="0.5"/>
        <param name="defaultEstimatorType" type="Value::Int1" x="2"/>
</section>
```

DeepFake estimation mode.

Currently, available values for selecting estimation scenario are: Default, M1 and M2. The scenario `Default` means the mode is specified in config file. @see ISettingsProvider.

Implementation description:

DeepFake Estimation Mode Enum

```
enum class DeepFakeMode {
   Default,  // Specified in config file.
   M1,       // M1.
   M2        // M2.
};
```

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 23: "Thresholds for `MedicalMaskEstimation`"

Parameter	Description	Type	Threshold range	Default threshold	FPR range	TPR range
mask	range [0..1]	"Value:: Float1"	0.65 – 0.9	0.65	0.014 – 0.01	0.976 – 0.886
noMask	range [0..1]	"Value:: Float1"	0.65 – 0.79	0.65	0.01 – 0.005	0.94 – 0.903
occludedFace	range [0..1]	"Value:: Float1"	0.5 – 0.602	0.5	0.016 – 0.01	0.924 – 0.881

Table 24: "Thresholds for `MedicalMaskEstimationExtended`"

Parameter	Description	Type	Threshold range	Default threshold	FPR range	TPR range
maskExtended	range [0..1]	"Value:: Float1"	0.65 – 0.784	0.65	0.013 – 0.01	0.923 – 0.894
noMaskExtended	range [0..1]	"Value:: Float1"	0.65 – 0.79	0.65	0.01 – 0.005	0.94 – 0.903
maskNotInPlaceExten	range [0..1]	"Value:: Float1"	0.65 – 0.85	0.65	0.009 – 0.005	0.918 – 0.833
occludedFaceExtended	range [0..1]	"Value:: Float1"	0.5 – 0.602	0.5	0.016 – 0.01	0.924 – 0.881

Example:

```
<section name="MedicalMaskEstimatorV3::Settings">
        <param name="maskExtendedThreshold" type="Value::Float1" x="0.65"/>
```

```
        <param name="noMaskExtendedThreshold" type="Value::Float1" x="0.65"
            />
        <param name="maskNotInPlaceExtendedThreshold" type="Value::Float1" x
            ="0.65"/>
        <param name="occludedFaceExtendedThreshold" type="Value::Float1" x="
            0.5"/>
        <param name="maskThreshold" type="Value::Float1" x="0.65"/>
        <param name="noMaskThreshold" type="Value::Float1" x="0.65"/>
        <param name="occludedFaceThreshold" type="Value::Float1" x="0.65"/>
    </section>
```

RedEyeEstimator settings

Red eye estimator evaluates whether person's eyes are red in a photo or not. Red eye estimation depends on threshold value listed below. These threshold value set to optimal by default.

Parameter	Description	Type	Default value
redEyeThreshold	redEyeThreshold threshold in [0..1] range.	"Value::Float1"	0.5

Example:

```
<section name="RedEyeEstimator::Settings">
        <param name="redEyeThreshold" type="Value::Float1" x="0.5"/>
</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 settings

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

Liveness protects from `presentation attacks` - when user tries to cheat biometric system by demonstrating fake face to the face capturing camera, but not from `image substitution attacks` - when fake image is sent directly to the system, bypassing the camera.

`LivenessOneShotRGBEstimator` supports images, which are captured on Mobile devices or Webcam (PC or laptop). Correct working of the estimator with other source images is not guaranteed.

Supported shooting mode: cooperative, which means that user must interact with the camera and look at it.

User scenarios examples: authentication in mobile application, confirmation of transactions with biometric facial verification.

Image resolution minimum requirements:

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

Parameter	Description	Type	Default value
useMobileNet	use mobile version	"Value::Int1"	0
realThreshold	threshold in [0..1] range.	"Value::Float1"	0.5
qualityThreshold	threshold in [0..1] range.	"Value::Float1"	0.5
calibrationCoeff	coefficient in [0..1] range.	"Value::Float1"	0.8
mobileRealThreshold	threshold in [0..1] range.	"Value::Float1"	0.5
mobileQualityThreshold	threshold in [0..1] range.	"Value::Float1"	0.5
mobileCalibrationCoeff	coefficient in [0..1] range.	"Value::Float1"	0.77

```
<section name="LivenessOneShotRGBEstimator::Settings">
    <param name="useMobileNet" type="Value::Int1" x="0" />
    <!--Parameters for backend version (useMobileNet == 0) -->
```

```
        <param name="realThreshold" type="Value::Float1" x="0.5"/>
        <param name="qualityThreshold" type="Value::Float1" x="0.5" />
        <param name="calibrationCoeff" type="Value::Float1" x="0.8"/>
        <!--Parameters for mobile version (useMobileNet == 1) -->
        <param name="mobileRealThreshold" type="Value::Float1" x="0.5"/>
        <param name="mobileQualityThreshold" type="Value::Float1" x="0.5" />
        <param name="mobileCalibrationCoeff" type="Value::Float1" x="0.77"/>
    </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>
```

Natural Light Estimator settings

Natural Light estimator is trained to predict natural of light on the face image.

It does so by returning a score value between [0;1] which will be closer to 1 if a light is more likely to be natural 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.

NaturalLight 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 NaturalLightEstimator::Settings section:

Parameter	Description	Type	Default value
naturalLightThreshold	threshold	"Value::Float1"	0.5

Example:

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

BlackWhite Estimator settings

Estimator checks if image is color, grayscale or infrared.

Estimator accuracy depends on thresholds listed below.

Parameter	Description	Type	Default value
colorThreshold	threshold in [0..1] range	"Value::Float1"	0.5
irThreshold	threshold in [0..1] range.	"Value::Float1"	0.5

Estimator outputs `ImageColorEstimation` which consists of 2 scores and color image type as enum with possible values: Color, Grayscale, Infrared.

- For color image score `colorScore` will be close to 1.0 and the second one `infraredScore` - to 0.0;
- for infrared image score colorScore will be close to 0.0 and the second one`infraredScore` - to 1.0;
- for grayscale images both of scores will be near 0.0.

So `colorThreshold` is responsible for separating Color and Grayscale images; `irThreshold` is responsible for separating Infrared and Grayscale images.

```
<section name="BlackWhiteEstimator::Settings">
        <param name="colorThreshold" type="Value::Float1" x="0.5"/>
        <param name="irThreshold" type="Value::Float1" x="0.5"/>
</section>
```

Fish Eye Estimator settings

Fish Eye estimator is trained to predict fish eye effect on the face image.

It does so by returning a score value between [0;1] which will be closer to 1 if a fisheye effect is more likely to be applied to the image 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.

Fish Eye 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 FishEyeEstimator::Settings section:

Parameter	Description	Type	Default value
fishEyeThreshold	threshold	"Value::Float1"	0.5

Example:

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

Background Estimator settings

This estimator is designed to evaluate the background in the original image.

Estimator accuracy depends on the thresholds listed below. The scores are defined in [0,1] range. If two scores are above the threshold, then the background is solid, otherwise the background is not solid.

Parameter	Description	Type	Default value
backgroundThreshold	threshold in [0..1] range	"Value::Float1"	0.5
backgroundColorThreshold	threshold in [0..1] range	"Value::Float1"	0.3

```
<section name="BackgroundEstimator::Settings">
        <param name="backgroundThreshold" type="Value::Float1" x="0.5"/>
        <param name="backgroundColorThreshold" type="Value::Float1" x="0.3"
          />
</section>
```

Human Attribute Estimator settings

Human Attribute estimator is trained to predict a bunch of human attributes on the human image.

Human Attribute estimator sets outwear color bool values and age by comparing an output score with a corresponding threshold value listed in faceengine.conf file. User can modify this threshold in HumanAttributeEstimator::Settings section:

Parameter	Description	Type	Default value
blackUpperThreshold	threshold	"Value::Float1"	0.740
blueUpperThreshold	threshold	"Value::Float1"	0.655
brownUpperThreshold	threshold	"Value::Float1"	0.985
greenUpperThreshold	threshold	"Value::Float1"	0.700
greyUpperThreshold	threshold	"Value::Float1"	0.710
orangeUpperThreshold	threshold	"Value::Float1"	0.420
purpleUpperThreshold	threshold	"Value::Float1"	0.650
redUpperThreshold	threshold	"Value::Float1"	0.600
whiteUpperThreshold	threshold	"Value::Float1"	0.820
yellowUpperThreshold	threshold	"Value::Float1"	0.670
blackLowerThreshold	threshold	"Value::Float1"	0.700
blueLowerThreshold	threshold	"Value::Float1"	0.840
brownLowerThreshold	threshold	"Value::Float1"	0.850
greenLowerThreshold	threshold	"Value::Float1"	0.700
greyLowerThreshold	threshold	"Value::Float1"	0.690
orangeLowerThreshold	threshold	"Value::Float1"	0.760
purpleLowerThreshold	threshold	"Value::Float1"	0.890
redLowerThreshold	threshold	"Value::Float1"	0.600
whiteLowerThreshold	threshold	"Value::Float1"	0.540
yellowLowerThreshold	threshold	"Value::Float1"	0.930
adultThreshold	threshold	"Value::Float1"	0.940

Example:

```xml
<section name="HumanAttributeEstimator::Settings">
                <param name="blackUpperThreshold" type="Value::Float1" x="
                    0.740"/>
                <param name="blueUpperThreshold" type="Value::Float1" x="
                    0.655"/>
                <param name="brownUpperThreshold" type="Value::Float1" x="
                    0.985"/>
                <param name="greenUpperThreshold" type="Value::Float1" x="
                    0.700"/>
                <param name="greyUpperThreshold" type="Value::Float1" x="
                    0.710"/>
                <param name="orangeUpperThreshold" type="Value::Float1" x="
                    0.420"/>
                <param name="purpleUpperThreshold" type="Value::Float1" x="
                    0.650"/>
                <param name="redUpperThreshold" type="Value::Float1" x="
                    0.600"/>
                <param name="whiteUpperThreshold" type="Value::Float1" x="
                    0.820"/>
                <param name="yellowUpperThreshold" type="Value::Float1" x="
                    0.670"/>

                <param name="blackLowerThreshold" type="Value::Float1" x="
                    0.700"/>
                <param name="blueLowerThreshold" type="Value::Float1" x="
                    0.840"/>
                <param name="brownLowerThreshold" type="Value::Float1" x="
                    0.850"/>
                <param name="greenLowerThreshold" type="Value::Float1" x="
                    0.700"/>
                <param name="greyLowerThreshold" type="Value::Float1" x="
                    0.690"/>
                <param name="orangeLowerThreshold" type="Value::Float1" x="
                    0.760"/>
                <param name="purpleLowerThreshold" type="Value::Float1" x="
                    0.890"/>
                <param name="redLowerThreshold" type="Value::Float1" x="
                    0.600"/>
                <param name="whiteLowerThreshold" type="Value::Float1" x="
                    0.540"/>
                <param name="yellowLowerThreshold" type="Value::Float1" x="
                    0.930"/>

                <param name="adultThreshold" type="Value::Float1" x="0.940"
```

```
            />
        </section>
```

Portrait Style Estimator settings

This estimator is designed to evaluate the status of a person's shoulders in the original image.

Estimator accuracy depends on the threshold listed below.

Parameter	Description	Type	Default value
notPortraitStyleThreshold	threshold in [0..1] range	"Value::Float1"	0.2
portraitStyleThreshold	threshold in [0..1] range	"Value::Float1"	0.35
hiddenShouldersThreshold	threshold in [0..1] range	"Value::Float1"	0.2

```
<section name="PortraitStyleEstimator::Settings">
        <param name="notPortraitStyleThreshold" type="Value::Float1" x="0.2"
            />
        <param name="portraitStyleThreshold" type="Value::Float1" x="0.35"/>
        <param name="hiddenShouldersThreshold" type="Value::Float1" x="0.2"
            />
</section>
```

HumanFace detector settings

Parameter	Description	Type	Default value
humanThreshold	Human detection score threshold in [0..1] range.	"Value::Float1"	0.5
nmsHumanThreshold	Human overlap threshold for NMS in [0..1] range	"Value::Float1"	0.4
faceThreshold	Face detection score threshold in [0..1] range.	"Value::Float1"	0.5
nmsFaceThreshold	Face overlap threshold for NMS in [0..1] range	"Value::Float1"	0.3
associationThreshold	Association score threshold in [0..1] range.	"Value::Float1"	0.5
minFaceSize	Minimum face size in pixels.	"Value::Int1"	50
batchCapacity	Non-public parameter. Do not change.	"Value::Int1"	8
cropPaddingAlignment	Non-public parameter. Do not change.	"Value::Int1"	64

```
<section name="HumanFaceDetector::Settings">
    <param name="humanThreshold" type="Value::Float1" x="0.5"/>
    <param name="nmsHumanThreshold" type="Value::Float1" x="0.4"/>
    <param name="faceThreshold" type="Value::Float1" x="0.5"/>
    <param name="nmsFaceThreshold" type="Value::Float1" x="0.3"/>
    <param name="associationThreshold" type="Value::Float1" x="0.5"/>
    <param name="minFaceSize" type="Value::Int1" x="50"/>
    <param name="cropPaddingAlignment" type="Value::Int1" x="64" />
    <param name="batchCapacity" type="Value::Int1" x="8" />
</section>
```

Landmarks detector settings

Parameter	Description	Type	Default value
useLNet	To detect Landmarks68	"Value::Int1"	1
useSLNet	To detect Landmarks5	"Value::Int1"	1

```
<section name="LandmarksDetector::Settings">
        <param name="useLNet" type="Value::Int1" x="1" />
        <param name="useSLNet" type="Value::Int1" x="1" />
</section>
```

Note Please pay attention, both parameters cannot be disabled at the same time. In this case, you will receive the error code (`fsdk::FSDKError::InvalidConfig`), and logs like below:

```
[30.08.2022 15:47:15] [Error] [FaceLandmarksDetector] Failed to create
    FaceLandmarksDetector! The both parameters: "useSLNet" and "useLNet" in
    section "LandmarksDetector::Settings" are disabled at the same time.
```

Crowd estimator settings

Parameter	Description	Type	Default value
defaultEstimatorType	Type of the estimator	`"Value:: String"`	TwoNets
minHeadSize	Target minHeadSize	`"Value::Int1 "`	6
batchCapacity	Non-public parameter. Do not change.	`"Value::Int1 "`	1
cropPaddingAlignment	Non-public parameter. Do not change.	`"Value::Int1 "`	0
lowThreshold	Non-public parameter. Do not change.	`"Value:: Float1"`	0.1
upThreshold	Non-public parameter. Do not change.	`"Value:: Float1"`	100

There are next possible values for the `defaultEstimatorType` parameter:

- Single - working mode with one network usage

- TwoNets - working mode with two networks usage

```
<section name="CrowdEstimator::Settings">
        <!-- Available types are: Single, TwoNets -->
        <param name="defaultEstimatorType" type="Value::String" text="
          TwoNets"/>
        <param name="minHeadSize" type="Value::Int1" x="6"/>
        <param name="batchCapacity" type="Value::Int1" x="1"/>
        <param name="cropPaddingAlignment" type="Value::Int1" x="0"/>
        <param name="lowThreshold" type="Value::Float1" x="0.1"/>
        <param name="upThreshold" type="Value::Float1" x="100"/>
</section>
```

LivenessDepthRGBEstimator settings

LivenessDepthRGBEstimator estimator performs liveness check via pair of depth image and RGB 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"	0.5

```xml
<section name="LivenessDepthRGBEstimator::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="0.5"/>
</section>
```

FightsEstimator settings

FightsEstimator estimator performs a fight detection on the several frame sequences from the target video. It exposes different parameters where each one of them let you configure estimator for your specific use case.

Parameter	Description	Type	Default value
batchSize	count of frames in one sequence (batch)	"Value::Int1"	5
minBatchCount	minimum sequences (batches) count	"Value::Int1"	5
cropSize	internal crop size. do not change it!	"Value::Int1"	224
threshold	score threshold above which the video is considered to contain a fight. Threshold in [0..1] range.	"Value::Float1"	0.5
scoreNorm	normalization parameter. do not change it!	"Value::Float1"	1.8

```
<section name="FightsEstimator::Settings">
        <param name="batchSize" type="Value::Int1" text="5"/>
        <param name="minBatchCount" type="Value::Int1" x="5"/>
        <param name="cropSize" type="Value::Int1" x="225"/>
        <param name="threshold" type="Value::Float1" x="0.5"/>
        <param name="scoreNorm" type="Value::Float1" x="1.8"/>
</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	Type	Default value
cpuClass	"Value::String"	"auto"
deviceClass	"Value::String"	"cpu"
numThreads	"Value::Int1"	-1
verboseLogging	"Value::Int1"	0
numComputeStreams	"Value::Int1"	4
programCacheSize	"Value::Int1"	128
defaultGpuDevice	"Value::Int1"	0

Parameters description:

cpuClass - class of cpu by supported instructions - cpu, sse4, avx, avx2, arm, auto.

deviceClass - execution device type - cpu, gpu.

numThreads - number of worker threads. Default: number of CPU logical cores.

verboseLogging - level of log verbosity. 1 - Errors, 2 - Warnings, 3 - Info, 4 - Debug.

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.

programCacheSize - maximum number of Program objects in cache. Should be less than 10000.

defaultGpuDevice - default GPU device number.

> 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:

```xml
<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`.