

Configuration Guide

written for LUNA SDK Mobile Android version 5.8.0

Contents

- Configuration Guide** **3**
- Configuration file location 3

- Settings** **4**
- System settings 4
- Descriptor factory settings 5
- FaceDetV2 detector settings 6
- LNet 7
- HeadPoseEstimator settings 8
- EyeEstimator settings 9
- Medical mask estimator settings 10
- Quality estimator settings 12
- GlassesEstimator settings 13
- LivenessOneShotRGBEstimator settings 14

- Runtime settings** **16**

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.

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:

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 |

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" />  
</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: 59 | "Value::Int1" | 59 |
| useMobileNet | MobileNet is faster but less accurate. Possible values: 0 - don't use mobile net version, 1 - use mobile net version. | "Value::Int1" | 1 |
| distance | Distance between descriptors on matching. L1 faster, L2 make better precision. Possible values: L1, L2. Model 59 supports just L2 distance. | "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 |

Example:

```
<section name="DescriptorFactory::Settings">
  <param name="model" type="Value::Int1" x="59" />
  <param name="useMobileNet" type="Value::Int1" x="1" />
  <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>
```

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.

HeadPoseEstimator settings

In mobile mode, HeadPose estimator is able to compute head pose angles using raw input image data only.

Default configuration settings enables estimation method by image.

| Parameter | Type | Default value |
|--------------------------|---------------|---------------|
| useEstimationByImage | "Value::Int1" | 1 |
| useEstimationByLandmarks | "Value::Int1" | 0 |

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 | "Value::Int1" | 1 |

Example:

```
<section name="EyeEstimator::Settings">  
  <param name="useStatusPlan" type="Value::Int1" x="1"/>  
</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 6: "Thresholds for MedicalMaskEstimation"

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

Table 7: "Thresholds for MedicalMaskEstimationExtended"

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

Example:

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

```
<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>
```

Quality estimator settings

Quality estimator looks at several image parameters, like lightness (think overexposure), darkness (think underexposure), blurriness, illumination uniformity value, specularly 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>
```

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>
```

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" " | 0.94 |
| 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" " | 0.99 |

```
<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="0.94"/>
<!--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="0.99"/>
</section>
```

Runtime settings

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

Note: The 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.

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 |
|-------------------------------|--|-----------------|---------------|
| <code>cpuClass</code> | Class of cpu by supported instructions - cpu, arm, auto. | "Value::String" | "auto" |
| <code>deviceClass</code> | Execution device type - cpu, gpu. | "Value::String" | "cpu" |
| <code>numThreads</code> | Number of worker threads. Default: number of CPU logical cores. | "Value::Int1" | -1 |
| <code>verboseLogging</code> | Level of log verbosity. 1 - Errors, 2 - Warnings, 3 - Info, 4 - Debug. | "Value::Int1" | 0 |
| <code>programCacheSize</code> | Maximum number of Program objects in cache. Should be less than 10000. | "Value::Int1" | 128 |

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.

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="programCacheSize" type="Value::Int1" x="128" />
</section>
```