

Configuration Guide

written for LUNA SDK Mobile iOS version 4.6.0

Contents

LUNA SDK Configuration Guide	3
Configuration file location	3
Settings	4
System settings	4
Descriptor factory settings	5
FaceDetV2 detector settings	6
LNet	7
HeadPoseEstimator settings	7
BestShotQualityEstimator settings	8
EyeEstimator settings	9
Runtime settings	10
Licensing	11
Licensing settings	11

LUNA SDK 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.

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

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="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: 54	"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. Note: model 54 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

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>
```

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.

HeadPoseEstimator settings

In mobile mode HeadPose estimator is able to compute head pose angles in only one way using raw input image data. 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>
```

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>
```

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
cpuClass	Class of cpu by supported instructions - cpu, arm, auto.	"Value::String"	"auto"
deviceClass	Execution device type - cpu, gpu.	"Value::String"	"cpu"
numThreads	Number of worker threads. Default: number of CPU logical cores.	"Value::Int1"	-1
verboseLogging	Level of log verbosity. 1 - Errors, 2 - Warnings, 3 - Info, 4 - Debug.	"Value::Int1"	0

Note: 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.

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" />
</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>
```