VisionLabs LUNA SDK Licensing

Contents

1 Introduction

This document includes information about LUNA SDK licensing. This information is required for any engineer or developer who wants to utilize SDK features.

Here you can find:

- General information about LUNA SDK licensing;
- The list of licensed features;
- Online and offline license activation description.

2 General information

FaceEngine supports per-features node-locked licensing for every supported platform. This means that a final customer version of FaceEngine might be customized by activating and deactivating the licensed features. For that reason, not all algorithms and modules described in this book might be available to you.

Each SDK instance should be activated on every device. License, which was activated for one device, could not be used on some other device.

Interface for License objects ILicense (see file ILicense.h) gives you possibility to:

- check, if license is already activated;
- save license for next usage to some file;
- load license from file;
- check if some feature of FaceEngine is available for this license.

Typical usage scenario:

- Create a IFaceEngine object (see file FaceEngine.h);
- Get license pointer through `fsdk::IFaceEngine::getLicense`;
- Make activation for that license object through `fsdk::activateLicense`. This method requires full or relative path to the "license.conf" file.

3 Licensed features

To work with license features in code the *LicenseFeature* enum should be used (see file ILicense.h). Some features are not available for some platforms.

> Descriptor license feature enables you to create a maximum of 100000 descriptors on server platforms.

Full list of available features and their associated estimators:

- **Detection** - enables you to to find person face in a frame.

  Available instances:

  - IDetector
  - IOrientationEstimator

- **BestShot** - enables you to select bestshot from the provided images.

  Available instances:

  - IBestShotQualityEstimator
  - IQualityEstimator
  - IAGSEstimator (deprecated, use IBestShotQualityEstimator instead)
  - IHeadPoseEstimator (deprecated, use IBestShotQualityEstimator instead)

- **Attributes** - enables you to create estimate basic attributes.

  Available instances:

  - IAttributeEstimator
  - IChildEstimator
  - ICredibilityCheckEstimator

- **Emotions** - enables you to estimate emotions.

  Available instances:

  - IEmotionsEstimator
  - IMouthEstimator

- **FaceFeatures** - enables you to estimate face features.

  Available instances:

  - IGlassesEstimator
  - IEyeEstimator
  - IGazeEstimator
  - IOverlapEstimator

- **Liveness** - enables you to use Liveness features.

  Available instances:

  - ILivenessRGBMEstimator
  - ILivenessDepthEstimator
  - ILivenessFlowEstimator
  - ILivenessIREstimator
  - IHeadPoseAndShouldersLivenessEstimator
  - ILivenessFlyingFacesEstimator

- **Descriptor** - enables you to work with descriptors.

  Available instances:

  - IDescriptor
  - IDescriptorBatch
  - IDescriptorExtractor
  - IDescriptorMatcher

- **DescriptorIndex** - enables you to create descriptors index.

  Available instances:

  - IIndexBuilder
  - IDenseIndex
  - IDynamicIndex

- **HumanDetection** - enables you to perform human bodies detection.

  Available instances:

  - IHumanDetector

- **LivenessEngine** - enables you to check if there is a real person or not in a video. See "LivenessEngine_Handbook.pdf" for details.

  Available instances:

  - ILivenessEngine

- **TrackEngine** - enables you to track a person in a video. See "TrackEngine_Handbook.pdf" for details.

  Available instances:

  - ITrackEngine

- **PPEDetection** - enables you to predicts wether a person is wearing one or multiple types of protection equipment.

  Available instances:

- IPPEEstimator
- IHeadWearEstimator

- **MobileLiveness** - enables you to predict whether the person's face is real or fake (photo, printed image).

  Available instances:

  - ILivenessOneShotRGBEstimator

- **MedicalMaskDetection** - enables you to detect a medical face mask on the face in the source image.

  Available instances:

  - IMedicalMaskEstimator

- **ReIdDescriptor** - enables you to work with human descriptors (ReID).

  Available instances:

  - IDescriptor (with ReID versions)
  - IDescriptorBatch (with ReID versions)
  - IDescriptorExtractor (with ReID versions)
  - IDescriptorMatcher (with ReID versions)
  - ICrowdEstimator
  - IHumanFaceDetector
  - IHumanAttributeEstimator

- **ISOCheck** - enables you to work with ISO check estimators:

  Available instances:

  - IBackgroundEstimator
  - IBlackWhiteEstimator
  - IEyeBrowEstimator
  - IFishEyeEstimator
  - INaturalLightEstimator
  - IPortraitStyleEstimator
  - IDynamicRangeEstimator
  - IRedEyeEstimator

- **DepthRGBLiveness** - enables you to work with LivenessDepthRGBEstimator:

  Available instances:

  - ILivenessDepthRGBEstimator

- **FightsEstimation** - enables you to work with FightsEstimator:

  Available instances:

- IFightsEstimator

- **BodyAttributes** - enables you to make a human body attributes estimation Available instances:

    - IHumanAttributeEstimatorPtr

- **NIRLiveness** - enables you to make a NIR Liveness estimation Available instances:

    - INIRLivenessEstimator

If the license for the selected feature is invalid, the factory instantiation method will return result with error code and empty object.

See FeatureMap.htm for additional information about these features.

For example, method: *IFaceEngine::createAGSEstimator* will return *ResultValue* with *FSDKError::LicenseError* empty *IAGSEstimatorPtr* object in case if LicenseFeature::BestShot in not available.

4 License activation

This section describes license activation.

License activation should be performed after the distribution package is unarchived.

> The generated license can only be used on the device where it was activated as the fingerprint of the device is used for its creation.

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.

4.1 Online activation

Online activation is performed when you have an Internet connection on the device where the LUNA SDK license is activated.

Licensing configuration options are specified via the "license.conf" file which is an XML document with special tag formatting.

> You can find the "license.conf" file in the "data" directory in the distribution package.

This file is mandatory for license activation. You must fill it with the correct values before launching LUNA SDK.

> Fill in this file carefully. Incorrectly entered data may cause problems with activation on the device.

4.1.1 Online activation process

The activation process is as follows:

- Request the **Server**, **EID**, and **ProductID** from VisionLabs
- Go to the "data" directory
- Open the "license.conf" file
- Enter the received parameters
- Save changes in the "license.conf" file

The license key will be generated and saved to the "data" directory.

Now you can use LUNA SDK.

4.2 Offline activation

Offline activation is performed when you do not have an Internet connection on the device where the LUNA SDK license is activated.

In this case, you should create a fingerprint of your device and use it to receive a license key on any other device with the Internet.

4.2.1 Offline activation process

The activation process is as follows:

- Request the website address for license key activation and EID from VisionLabs

  Perform the following steps on the device where the license should be activated:

- Go to the "data" directory

- Open the "license.conf" file

- Enter the received EID

- Save changes in the "license.conf" file

- Go to the "./bin"

- Run the "FingerprintViewer" utility to create a fingerprint of your device:

  – For Windows: launch the "FingerprintViewer.exe"

  – For Linux:

    * Give execution permissions to the "FingerprintViewer" utility

    ```
    chmod +x FingerprintViewer
    ```

    * Run the utility

    ```
    ./FingerprintViewer
    ```

- The fingerprint will be printed in the console. Copy and save it.

  Perform the following steps on the device with the Internet:

- Go to the website to receive license (the website address was received on the first step of this instruction)

- Enter your EID to enter the website, and using your device fingerprint activate a license.

- Download license certificate. Please pay attention, by default filename is "licenseFile.v2c". Below are the steps (please choose one of them), how to configure. Move the "licenseFile.v2c" file to your device "data" directory of LUNA SDK.

  – Modify param `"Filename"` in file "license.conf" as in example below:

    ```
    <param name="Filename" type="Value::String" text="licenseFile.v2c"/>
    ```

  – Rename "licenseFile.v2c" to "license.dat". Param `"Filename"` in file "license.conf" **need no modifications**, by default as below:

```
<param name="Filename" type="Value::String" text="license.dat"/>
```

Perform the following step on the device where the license should be activated:

- Copy the received license key "license.dat" to the "data" directory of LUNA SDK.

4.3 License file parameters description

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"	license.dat
ActivationMode	Activation Mode	"Value::String"	Hardware
ConnectionTimeout	Request timeout (in seconds)	"Value::Int1"	15
ServerRetriesCount	Retry attempts count	"Value::Int1"	3

Server, **EID**, and **ProductID** - this information must be requested from VisionLabs and written to the file. It is mandatory for activation procedure.

EID field should be filled in for the online and offline activation.

Server and **ProductID** fields should be filled in for the online activation only.

Filename - the name of the file to save license after activation. The maximum length of the setting string is 64 symbols. Do not change this name!

ActivationMode - activation mode specifies which path to choose when retrieving device UID during license activation stage. Possible values are: Hardware, JNI.

ConnectionTimeout - set the maximum time in seconds that you allow to transfer operation to take. Normally, name lookups can take a considerable time and limiting operations risk aborting perfectly normal operations. Timeout 0 (zero) means it never times out during transfer. The ConnectionTimeout can't be set to a negative value, or a value that is large than maximal, which is 300 seconds.

ServerRetriesCount - enables an application to handle transient failures when it tries to connect to a service or network resource, by transparently retrying a failed operation. This can improve the stability of the application. The ServerRetriesCount can't be set to a negative value, or a value that is large than maximal, which is 100 attempts.

By default, the file is created in the "data" directory. The file has a binary format. At the next launch of the product on the same device, a license will be read from this file.

An example of the "license.conf" file:

```xml
<?xml version="1.0"?>
<settings>
<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="license.dat"/>
    <param name="ActivationMode" type="Value::String" text="Hardware"/>
    <param name="ConnectionTimeout" type="Value::Int1" x="15"/>
    <param name="ServerRetriesCount" type="Value::Int1" x="3"/>
</section>
</settings>
```