



VisionLabs
MACHINES CAN SEE

Quick Start Guide

Contents

Introduction	3
Overview	3
FaceEngine	3
TrackEngine	3
Getting started	6
Bundled face descriptor model versions	7
Bundled reidentification descriptor model version	8
Where is the code	9

Introduction

This short guide will help you to get started with the SDK.

Overview

Luna SDK contains two primary libraries:

- **FaceEngine SDK** is the core library. It is described in the [FaceEngine_Handbook.pdf](#);
- **LivenessEngine SDK** includes complex liveness checks. It is described in the [LivenessEngine_Handbook.pdf](#).

The **TrackEngine** tool is used for receiving images from video.

See [Licensing.pdf](#) for information about features licensing and license activation.

See [FeatureMap.htm](#) for additional information about available functionality and features.

The [ConfigurationGuide.pdf](#) includes information about SDK configurations.

FaceEngine

FaceEngine SDK enables you to implement the following functionality:

- Faces and human body detection in still images;
- Landmarks location on faces and human bodies;
- Faces warping;
- Human body resize;
- Face and human descriptor extraction and matching;
- Descriptor search index creation (fast Hierarchical Navigable World graph or HNSW shortly);
- Batching for processing performance improvements.

FaceEngineSDK library is available in two versions: Frontend Edition (FE) and Complete Edition (CE):

- Complete Edition supports all functionality;
- Frontend Edition is a light version with excluded descriptor functionality (does not support descriptor extraction and matching).

LivenessEngineSDK library is designed to distinguish a real user's face from a fake one (e.g., printed on paper or displayed on the screen). LivenessEngine uses FaceEngineSDK library modules and performs detection, warping, and various estimations.

TrackEngine

It is a tool for face detection and tracking on multiple sources. It allows to pick the most suitable still images for facial recognition from a sequence of video frames.

Note, that TrackEngine itself does not perform any facial recognition. It's purpose is to prepare required data for external systems.

SDK package structure

Folder	Contents
/include	C++ header files
/lib/<compiler>/<bitness>	Libraries (.lib on Windows, .so on Linux)
/bin/<compiler>/<bitness>	Binaries and tools (also .dll files on Windows)
/doc	Documentation
/data	Algorithm model data required to use the SDK
/examples	Sample code

Note 1: <compiler> is “gcc4” on Linux and “vs2015” on Windows.

Note 2: GCC versions 5 and later break backward compatibility with GCC4, so do not mix them.

Note 3: <bitness> is “x64” for 64-bit OS and “x86” for 32-bit OS.

Getting started

It is recommended to familiarize with the common FaceEngine SDK concepts and terminology first. For that, please refer to the handbook in “FaceEngine_Handbook.pdf”. Handbook contents is as follows:

- Chapters 1-3 cover common concepts and FaceEngine SDK modules;
- Chapters 4-7 describe each module in depth;
- Chapter 8 tells more about system requirements;
- Chapter 9 is dedicated to hardware and software requirements. Please make sure your system meets the requirements listed in chapter 9 before proceeding.

In appendixes one may find performance evaluation results and answers to some frequently asked questions.

Note, that the purpose of the handbook is to describe common concepts and give an idea what LUNA SDK is capable of. For detailed descriptions of particular functions, refer to the reference manual instead. FaceEngine reference manual (doxygen) is available in the **doc/fsdk/html/** directory.

FaceEngine SDK configuration parameters are described in **doc/ConfigurationGuide.pdf**.

Liveness SDK reference manual (doxygen) is available in the **doc/lSDK/html/** directory.

For additional information on Liveness SDK refer to the liveness handbook in **doc/LivenessEngine_Handbook.pdf**.

Document **doc/Featuremap.htm** contains information about matching between SDK facilities and data files which these facilities needs.

Bundled face descriptor model versions

The SDK implements several face descriptor models.

Multiple models are maintained for backward compatibility reasons, as cross-model matching is not supported.

You may find additional details on descriptors in section “Descriptor processing facility” and “Appendix A. Specifications” of the SDK Handbook.

Bundled reidentification descriptor model version

Three models are currently presented - the tracker is the 102 version of the descriptor model (used for tracking people, the light and fast version) and the precise one is the 103 version of the descriptor model (accurate, heavy and relatively slow), the regular is the 104 version of the descriptor model.

To create descriptors, batchs, matcher and extractor of human type you must pass version of human descriptor.

Where is the code

The example code for server and PC versions is located in the **/examples** folder. Please refer to doc/ExamplesGuide.pdf, for additional information.