id3 Face SDK 9.0.4

Cross-platform face recognition library

Developer's Guide

Document number: CJ388

Date: 2022/03/15

Legal Information

Contact information

For more information, please visit: https://www.id3.eu

Contents

List of Tables

List of Figures

1 Introduction

1.1 Scope

This document is the Developer's Guide of id3 Face SDK 9.0.4.

1.2 Terms and Definitions

For the purpose of this document, the following terms and definitions apply.

Algorithm
A sequence of instructions that tell a biometric system how to solve a particular problem. An algorithm will have a finite number of steps and is typically used by the biometric engine (i.e. the biometric system software) to compute whether a biometric sample and template match.

Biometric data
Data encoding a feature or features used in biometric verification.

Comparison
The process of comparing a biometric sample with a previously stored reference template or templates.

Comparison score
Numerical value resulting from a comparison.

Enrollment
The process of collecting biometric samples from a person and the subsequent preparation and storage of biometric reference templates representing that person's identity.

Extraction
The process of converting a captured biometric sample into biometric data so that it can be compared to a reference template; sometimes called 'characterization'.

False match
Comparison decision of 'match' for a biometric probe and a biometric reference that are from different biometric capture subjects.

False match rate
Expected probability that a captured biometric sample will be falsely declared to match to a single, randomly-selected, non-self biometric reference.

False non-match
Comparison decision of 'non-match' for a biometric probe and a biometric reference that are from the same biometric capture subject and of the same biometric characteristic.

False non-match rate
Proportion of the completed biometric mated comparison trials that result in a false non-match.

Match / Matching
The process of comparing a biometric sample against a previously stored template and scoring the level of similarity.

One-to-many search

Process in which a biometric probe of one biometric data subject is searched against the biometric references of more than one biometric data subject to return a candidate list or a comparison decision.

One-to-one comparison
Process in which a biometric probe from one biometric data subject is compared to a biometric reference from one biometric data subject to produce a comparison score.

Template / **Reference template**
Data, which represents the biometric measurement of an enrollee, used by a biometric system for comparison against subsequently submitted biometric samples.
NOTE -– this term is not restricted to mean only data used in any particular recognition method, such as template matching.

1.3 Abbreviated Terms

For the purpose of this document, the following abbreviations apply.

API	Application Programming Interface
ANSI	American National Standards Institute
GPU	Graphical Processing Unit
ICAO	International Civil Aviation Organization
IOD	Interocular distance
ISO	International Organization for Standardization
NIST	National Institute of Standards And Technology
ONVIF	Open Network Video Interface Forum
PAD	Presentation attack detection
SDK	Software Development Kit

2 Overview

id3 Face SDK is a cross-platform library aimed at system integrators willing to quickly add face detection, recognition, PAD and/or analysis capabilities to their products. It is available as a Software Development Kit (SDK) offering a comprehensive interface to simplify integration of the library on servers, desktops/laptops, mobile and edge devices.

2.1 Features & Benefits

id3 Face SDK offers the following features and benefits:

- Top performance facial biometrics, AI powered
- Optimized for AI hardware, including GPU acceleration, for fast operation
- Robust human face detection and tracking in digital images or video frames
- Small facial features template (268 bytes)
- Ultra-fast face template matching in one-to-one comparison and one-to-many search modes
- Multiple face analysis functionalities: landmarks estimation, pose estimation, mask detection, etc.
- Facial attributes determination for ICAO compliant portraits
- Face image quality assessment
- Accurate passive liveness detection methods to protect against biometric fraud, e.g. presentation attacks with photos and videos
- Compact library designed to run on most hardware configurations from high-end workstations to low-power edge devices
- Compatible with Windows, Linux, Android, MacOS and iOS operating systems
- Simple and comprehensive programming interface in various languages

2.2 Applications

- Public Safety
- Border Control
- Mobile apps
- ID Management
- Banking & Payment
- Automobile & Transportation
- Healthcare

2.3 Requirements

2.3.1 System Requirements

id3 Face SDK is compatible with the following platforms:

Processor Architectures	Operating Systems
Intel Core, Xeon (x86, x64)	Microsoft Windows 7, 8, 10 (32/64 bits), Microsoft Windows Server 2012 (64 bits), GNU/Linux with Kernel version >2.6.32 (64 bits), MacOS
arm64 M1	MacOS
ARMv7 (armeabi-v7a), ARMv8 (arm64-v8a)	Android 5.0+ (API >= 21)
ARMv7, ARMv7s, ARMv8	iOS 10.9+
NVIDIA Jetson Nano	JetPack 4.6
Raspberry Pi 4	Raspberry Pi OS armv7l

Table 2: System Requirements

For GPU usage, additional requirements are specified in section Additional Requirements for GPU Usage.

2.3.2 Memory requirements

The memory requirements depend mainly on the machine learning models loaded by the application. id3 Face SDK offers the possibility to load only the models that are required by the application.

2.3.3 Programming languages

id3 Face SDK provides API for the following programming languages:

- C
- C#
- Dart
- Java
- Swift

3 Technical Specifications

3.1 Technology overview

Thanks to many years of research and development in the field of computer vision and artificial intelligence, our experts have designed a unique algorithm that reproduces the visual recognition abilities of the human brain.

With the power of deep learning techniques trained on millions of faces, our technology outperforms human performance enabling unconstrained/non-voluntary real-time detection and recognition of faces in a crowd. It operates on any type of people face whatever the gender, age or race and is robust to intra-personal variations such as ageing, facial hair, scars/injuries, accessories (e.g. glasses, hats, etc.), cosmetics, etc. The technology is also robust to variations of the lightning conditions and has the ability to work with a large range of cameras under either visible or near-infrared light.

The face identification process is nearly instantaneous. It has the capability to compare millions of faces in less than one second on a single processing unit. The matching algorithm has also very low resource requirements enabling possible applications on secure elements.

id3 Technologies face recognition algorithm has proven excellent tradeoff between accuracy, speed and template size in the NIST ongoing Face Recognition Vendor Test (FRVT).

id3 Face SDK contains the following components:

- Face detector
- Face encoder
- Face matcher
- Face analyser
- Presentation attack detector (PAD)

3.2 Face detector

id3 Face SDK provides a face detection component used to detect and track faces in still images or video frames. It includes the following machine learning models:

	Model size
FaceDetector_3A	57,661 kB
FaceDetector_3B	984 kB

Table 3: Face detection models

Both models have the following characteristics:

Face angle tolerance	yaw < ±45° pitch < ±30° roll < 45°
Minimal IOD	10 pixels

Table 4: Face detection characteristics

NOTE — For optimum performance of the facial recognition, a minimal distance of 100 pixels between the eyes is recommended.

3.2.1 Face bounds

id3 Face SDK detects human faces and returns a rectangle around the detected faces, as shown in the figure below.

Figure 1: Face rectangle

3.2.2 Portrait bounds

id3 Face SDK detects human faces and returns a rectangle for ISO/ICAO compliant portraits, as shown in the figure below.

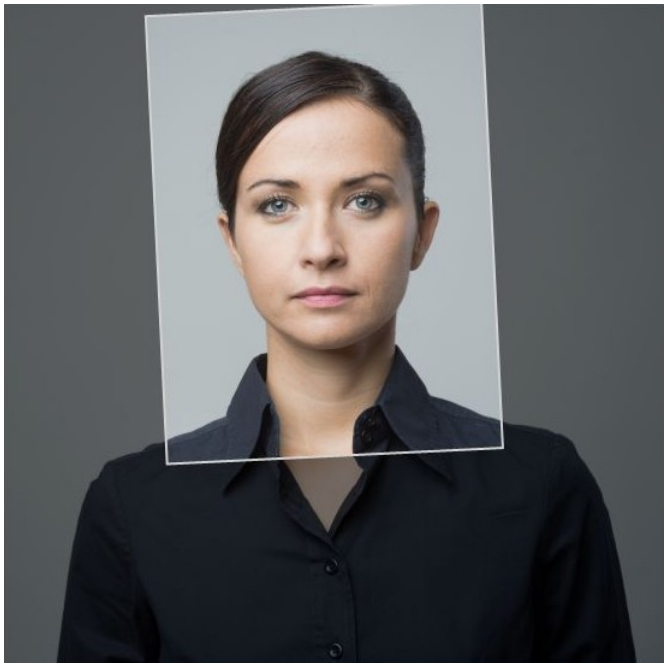

Figure 2: Face portrait rectangle

3.3 Face encoder

3.3.1 Templates

id3 Face SDK includes a component used to extract unique features from a detected face and encode them into a face template. It contains the following machine learning models:

	Template size	Model size
FaceEncoder_8A	268 bytes	152,457 kB
FaceEncoder_8B	140 bytes	2,123 kB
FaceEncoder_9A	264 bytes	127,365 kB
FaceEncoder_9B	140 bytes	1,996 kB

Table 5: Face encoding models

FaceEncoder_8A and **FaceEncoder8B** are only supported for previous projects compatibility. **FaceEncoder_8A** corresponds to the NIST FRVT submission id3_006.
FaceEncoder_9A provides the best possible accuracy with this SDK. It must be used for 1-to-N applications. It corresponds to the NIST FRVT submission id3_008.
FaceEncoder_9B provides a balance between accuracy, speed and model size. It is designed for 1-to-1 embedded applications. This algorithm is way less robust to pose variations than **FaceEncoder_9A**. It must be used in a controlled scenario.
Warning: Templates created with one encoder will not be compatible with ones created by another encoder.

3.3.2 Encoding quality

The component also provides a quality estimator that aims at predicting the power of matching of a created template. This functionality requires the model **FaceEncodingQualityEstimator_2A** to be loaded.

3.4 Face matcher

id3 Face SDK includes an ultra-fast face template matching algorithm which can operate in one-to-one or one-to-many mode. A decision threshold applied on the algorithm's output score determines the operational FMR (False Match Rate), i.e. the probability of a system to falsely accept faces (impostors). Since FMR and FNMR (False Non-Match Rate) are in inverse proportion to each other, FNMR will increase with higher decision thresholds.

Decision threshold

The decision threshold shall be defined according to the following table:

False Match Rate	Decision Threshold
1.10^{-2}	2000
5.10^{-3}	2300
2.10^{-3}	2700
1.10^{-3}	3000
5.10^{-4}	3300
2.10^{-4}	3700
1.10^{-4}	4000
5.10^{-5}	4300
2.10^{-5}	4700
1.10^{-5}	5000
5.10^{-6}	5300
2.10^{-6}	5700
1.10^{-6}	6000
5.10^{-7}	6300
2.10^{-7}	6700
1.10^{-7}	7000

Table 6: Decision Thresholds

3.5 Face analyser

id3 Face SDK can analyse a wide range of facial attributes. For instance, it can evaluate the compliancy of a portrait to the ISO/IEC 19794-5 full frontal and the ICAO Portrait Quality Standards thanks to the following modules:

- Landmarks

- Head pose
- Attributes
- Occlusions
- Expression
- Geometric attributes
- Photographic attributes
- Glasses attributes
- Background uniformity
- Eye openness
- Eye redness
- Eye gaze

Some other features are present in this analysis module:

- Age
- Mask

3.5.1 Landmarks

id3 Face SDK allows to locate the position of the facial landmarks including face, eye, eyebrow, lip and nose contours as shown in the figure below. Those landmarks are useful for other portrait quality features.
This functionality requires the model **FaceLandmarksEstimator2A** to be loaded.

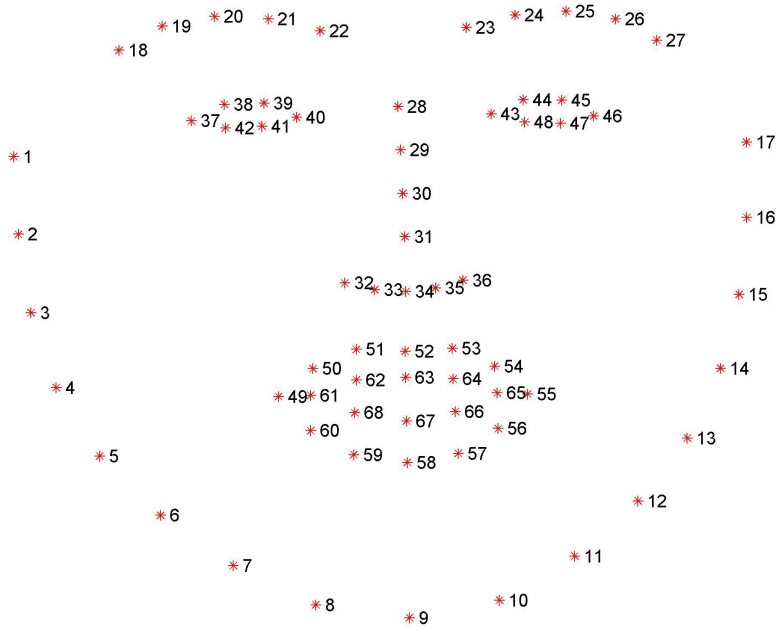

Figure 3: Face landmarks

3.5.2 Head pose

id3 Face SDK estimates the head pose parameters as described in the figure below.
This functionality requires the model **FacePoseEstimator2A** to be loaded.

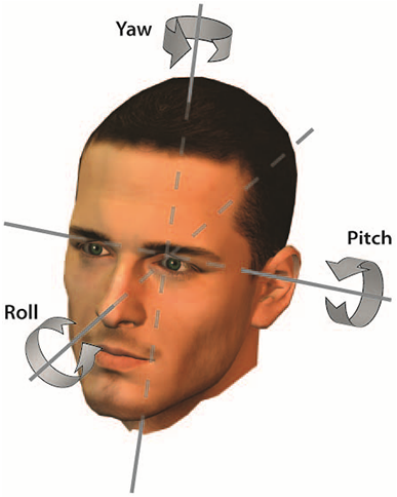

Figure 4: Head pose estimation (yaw, pitch, roll)

3.5.3 Attributes

id3 Face SDK allows to estimate the above facial attributes:

- **Glasses**: Estimates if the person is wearing glasses.
- **Hat**: Estimates if the person is wearing a hat.
- **Male**: Estimates if the person is male or female.
- **Mouth openess**: Estimates if the person's mouth is open.
- **Smile**: Estimates if the person is smiling.
- **Makeup**: Estimates if the person is wearing makeup.

This functionality requires the model **FaceAttributesClassifier2A** to be loaded.

3.5.4 Occlusions

id3 Face SDK allows to detect occlusions on the 4 strategic areas of a face (left eye, right eye, nose and mouth) as shown in the figure below.
This functionality requires the model **FaceOcclusionDetector1A** to be loaded.

3.5.5 Expression

id3 Face SDK allows to classify the facial expression among the following possibilities: angriness, disgust, fear, happiness, neutrality, sadness and surprise.
This functionality requires the model **FaceExpressionClassifier1A** to be loaded.

3.5.6 Geometric attributes

id3 Face SDK allows to compute the following geometric attributes of a detected face:

- **Resolution**: Checks if the image resolution is sufficient.

- **Vertical position**: Checks the distance from the bottom edge of the image to the imaginary line passing through the center of the eyes is between 50% - 70% of the total vertical length of the image.
- **Horizontal position**: Checks if the head is correctly centered horizontally.
- **Head image width ratio**: Checks if the image width is conform to the (image width / head width) ratio of 7:5. The head width is defined as the distance between the left and right ears.
- **Head image height ratio**: Checks if the distance between the base of the chin and the crown is less than 80% of the total height of the image.

This functionality does not require any model to be loaded.

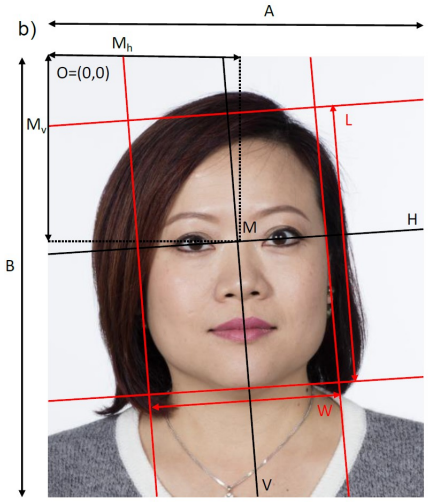

Figure 5: Geometric attributes

3.5.7 Photographic attributes

id3 Face SDK allows to compute the following photographic attributes of a detected face:

- **Sharpness**: Checks if the sharpness is sufficient.
- **Brightness**: Checks if the brightness is good.
- **Equal brigthness**: Checks if one side of the face is ligther than the other side.
- **Natural skin color**: Checks if the skin color is natural.
- **Contrast**: Checks if the contrast of the face is fine.
- **Pixelation**: Checks if the image is pixelized.
- **Flash reflection**: Checks if a flash reflection is present on the face.

This functionality does not require any model to be loaded.

3.5.8 Glasses attributes

id3 Face SDK allows to compute the following attributes of the glasses, if present:

- **Heavy frame**: Estimates if the glasses' frame is too heavy.
- **Frame on eye**: Estimates if the glasses frame is not on the eyes.
- **Tinted glasses**: Estimates if the glasses are tinted.

This functionality does not require any model to be loaded.

3.5.9 Background uniformity

id3 Face SDK allows to evaluate the uniformity of the background behind a face.
This functionality requires the model **FaceBackgroundUniformityEstimator1A** to be loaded.

3.5.10 Eye openness

id3 Face SDK allows to detect if a person's eyes are open or closed.
This functionality requires the model **EyeOpennessClassifier1A** to be loaded.

3.5.11 Eye redness

id3 Face SDK allows to detect if a person's eyes are red or not (due to flash mostly).
This functionality requires the model **EyeRednessClassifier1A** to be loaded.

3.5.12 Eye gaze

id3 Face SDK allows to estimate the looking direction (gaze) of a face.
This functionality requires the model **EyeGazeEstimator1A** to be loaded.

3.5.13 Age

id3 Face SDK allows to estimate the age of a person from his face.
This functionality requires the model **FaceAgeEstimator1A** to be loaded.

3.5.14 Mask

id3 Face SDK allows to detect if a person wears a face mask or not.
This functionality requires the model **FaceMaskClassifier1A** to be loaded.

Figure 6: Face mask detection

3.6 Presentation attack detector (PAD)

The FacePAD component allows to detect presentation attacks in still images or video frames.
It includes the following methods:

- Passive liveness method based on high-resolution color image analysis.

- Passive liveness method based on low-resolution color image analysis.
- Passive liveness method based on depth-map analysis.

NOTE – An active liveness method can also be implemented with this SDK using, for example, the head pose information or the smile.

3.6.1 Color-based PAD

The id3 Face SDK 9.0.4 provides various methods of PAD for color images:

- A classifier that infers the live or spoof decision based on a high-resolution image. This method requires the model file **FaceColorPad1A** to be loaded. It is required to use a face image with an IOD of at least 64 pixels for this method. This algorithm is mainly designed for mobile phones applications.
- A blurriness detector that allows to detect low-resolution attacks such as small ID card photos. This method requires the model **FaceBlurrinessDetector1A** to be loaded.
- A Moiré detector that allows to detect screen attacks. This method requires the model **FaceMoireDetector1A** to be loaded. It is recommended not to resize the input image to use this method since Moiré patterns are highly affected by interpolations.
- An attack support detector that allows to detect screen attacks and ID card attacks. This method requires the model **FaceAttackSupportDetector1A** to be loaded.

Each of those algorithms target various cases. They can each be used alone or be merged together using late fusion. If doing so, it is recommended to trigger an attack alert as soon as one of the methods outputs an attack decision. Otherwise, if none of the algorithms triggers an attack decision, then the presentation can be assumed to be bona-fide.

3.6.2 Depth-based PAD

This feature requires the model **FaceDepthPad2A** to be loaded. It also requires a depth camera. When using this PAD method, one must be careful about the range of validity of the depth information provided by the 3rd party sensor. It is recommended to use this PAD algorithm only in a valid range of depths. It is also required to use a face image with an IOD of at least 16 pixels for this method.

4 Biometric Performance

4.1 Face recognition

Factors affecting biometric accuracy

The overall accuracy of a facial recognition system may vary according to a number of factors such as:

- Quality of the camera system,
- Lighting conditions,
- Facial pose variations,
- Population under test,
- etc.

Performance metrics

False non match rate (FNMR) is the proportion of mated comparisons below a threshold set to achieve the false match rate (FMR) specified. FMR is the proportion of impostor comparisons at or above that threshold. Since FMR and FNMR is in inverse proportion to each other, choosing the operational threshold is a trade-off between system security and user convenience.

4.1.1 NIST FRVT evaluation

The Face Recognition Vendor Test (FRVT) was initiated by the National Institute of Standards and Technologies (NIST) in February 2017. It is aimed at measurement of the performance of automated face recognition technologies applied to a wide range of civil, law enforcement and homeland security applications including verification of visa images, de-duplication of passports, recognition across photojournalism images, and identification of child exploitation victims.

FRVT datasets

Visa images

- The images have geometry in reasonable conformance with the ISO/IEC 19794-5 Full Frontal image type. Pose is generally excellent.
- The images are of size 252x300 pixels. The mean interocular distance (IOD) is 69 pixels.
- The images are of subjects from greater than 100 countries, with significant imbalance due to visa issuance patterns.
- The images are of subjects of all ages, including children, again with imbalance due to visa issuance demand.
- Many of the images are live capture. A substantial number of the images are photographs of paper photographs.

Mugshot images

- The images have geometry in reasonable conformance with the ISO/IEC 19794-5 Full Frontal image type.
- The images are of variable sizes. The median IOD is 104 pixels. The mean IOD is 123 pixels.
- The images are of subjects from the United States.
- The images are of adults.
- The images are all live capture.

Wild images

- The images include many photojournalism-style images. Images are given to the algorithm using a variable but generally tight crop of the head. Resolution varies very widely. The images are very unconstrained, with wide yaw and pitch pose variation. Faces can be occluded, including hair and hands.
- The images are of adults.
- All of the images are live capture, none are scanned.

FRVT results

The latest report of the FRVT can be found here: https://pages.nist.gov/frvt/reports/11/frvt_11_report.pdf

The face encoder 9A of this SDK corresponds to the **id3_008** submission.
A complete report card can also be found here, showing among things, the evolution of our face recognition technology over the years: https://pages.nist.gov/frvt/reportcards/11/id3_008.html

5 Getting Started

5.1 SDK contents

5.1.1 Archive

The id3 Face SDK 9.0.4 is delivered as a compressed archive (ZIP file) that contains the following directories and files:

Directory	Description
`activation`	License activation
`bin`	Binaries
`dart`	Dart package
`docs`	Documentations
`dotnet`	.NET wrapper
`include`	Header files
`java`	Java archive

Table 7: Directory structure

5.1.2 Model files

The id3 Face SDK 9.0.4 requires model files to run. They are common to every platform and should be distributed according to the application's needs.

They can be downloaded at the following URL: https://cloud.id3.eu/index.php/s/y63PysbS3w6NPm6

Since there are lots of models in the folder, it is recommended to download and deploy only the necessary ones for the application. See the modules' documentation to know what models are required for each of them.

Note: Model files MUST NOT be renamed.

5.1.3 Sample applications

The id3 Face SDK 9.0.4 contains sample projects for the following programming languages:

- C/C++
- C#
- Dart
- Android (Kotlin application calling the Java API)

Those samples are no delivered in the ZIP archive. They can be found directly here: https://github.com/id3Technologies/face-sdk-samples
Samples demonstrate how to use the main functionalities of the id3 Face SDK 9.0.4, including:

- one-to-one face comparison
- face PAD
- face portrait analysis

5.2 License activation

To develop an application based on id3 Face SDK 9.0.4, an integrator should obtain an SDK license which includes all the modules and functionalities offered by the solution.

To complete the license activation procedure, you need:

- an activation key, or serial number, or id3 account credentials + product reference
- an internet connection

NOTE – Once the license is activated, the internet is no longer required.

During the activation procedure, the SDK license is bound to a uniquely identified ID named "HardwareID".

5.2.1 License Activation on desktop or server

To get your license file two ways are prefered:
First method is to use the command line activation software, for example on linux x64 :

- "cd activation/linux"
- "chmod +x id3LicenseActivationCLI"
- "./id3LicenseActivationCLI –help"

Second method is to directly use the **id3FaceLicense_Activate...()** APIs from the SDK inside your application.

5.2.2 License activation on Windows

On windows the graphical License Manager application may also be used to activate licenses. Please refer to the License Manager user's guide for more details.
Please note that on windows, and windows only, you can lock your license on a USB device.

5.2.3 License Activation on mobile devices

On mobile devices, depending of the hardware, each application you will develop, including samples, may require a different license file. Your applications must embed the call to the **id3FaceLicense_Activate...()** APIs at startup.
The mobile samples show how to retrieve the license file if you have been given a serial key or if you got an id3 account and product reference.

5.3 Binaries

Directory	Description
`bin\android\`	Android archive library
`bin\linux\armv8-jetson-nano`	Native libraries for NVIDIA Jetson Nano
`bin\linux\rpi4`	Native libraries for Raspberry Pi 4
`bin\linux\x64`	Native libraries for Linux x64
`bin\linux\x64_gpu`	Native libraries for Linux x64 with GPU support
`bin\windows\x64`	Native libraries for Windows x64
`bin\windows\x64_gpu`	Native libraries for Windows x64 with GPU support
`bin\windows\x86`	Native libraries for Windows x86

Table 8: Binaries

5.4 Documentations

The id3 Face SDK 9.0.4 comes with this Developer's Guide and the following documentation:

- C API Reference
- .NET API Reference
- Javadoc
- Dart package documentation

6 Application Programming Interface (API)

6.1 Overview

id3 Face SDK provides an easy to use programming interface available in various languages. The figure below provides an overview of the main components and objects:

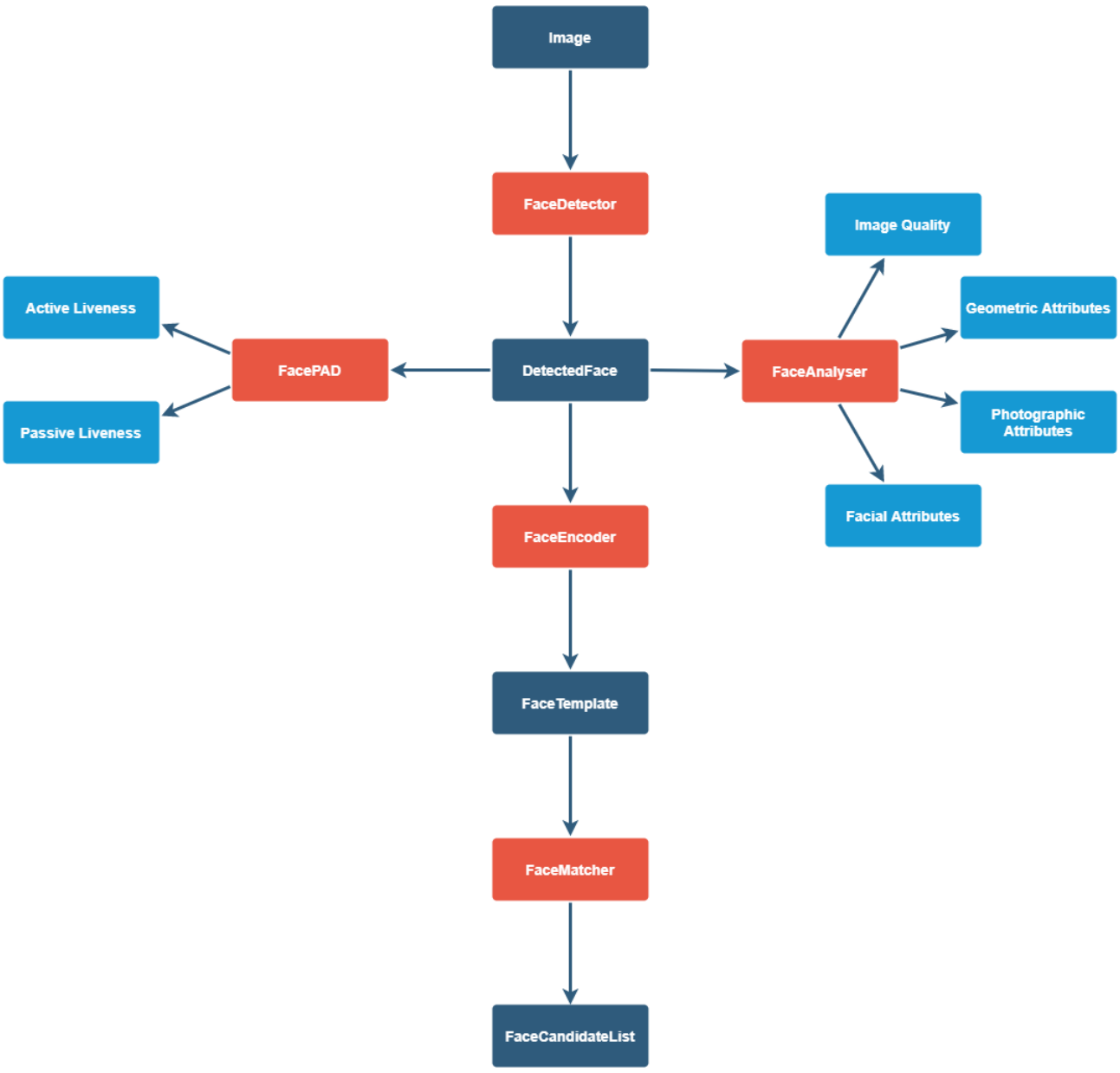

Figure 7: SDK Overview

6.2 Main Components

The id3 Face SDK 9.0.4 consists of several components providing easy access to advanced biometric functionalities. It includes:

Name	Description
`FaceAnalyser`	Provides a deep analysis of a portrait image features.
`FaceDetector`	Detects human faces in images and tracks them in consecutive images.
`FaceEncoder`	Extracts facial features and encodes face templates.
`FaceLibrary`	Manages machine learning models required to run the SDK.
`FaceLicense`	Activates and checks licenses.
`FaceMatcher`	Compares two face templates and searches a face template in a list.
`FacePAD`	Detects presentation attacks in still images or video frames using color or depth image.

Table 9: Main components

For detailed usage of each component please check each language API specifications.

7 Distribution

This section describes how to distribute your application implementing functionalities of the id3 Face SDK 9.0.4.

7.1 Distributing Windows applications

To distribute a Windows application that uses id3 Face SDK 9.0.4, you must deploy the following libraries and files:

- id3Face.dll
- Microsoft Visual C++ 2017 Redistributable
- Model files

7.2 Distributing Linux applications

To distribute a Linux application that uses id3 Face SDK 9.0.4, you must deploy the following libraries and files:

- libid3Face.so
- Model files

7.3 Additional Requirements for GPU Usage

On Windows or Linux

For Windows or Linux, GPU is supported only for the x86_64 architecture only. Please be aware that not all models support GPU acceleration.
Following additional dependencies are required, please refer to NVIDIA documentation for installation procedures relative to your operating system.

- Cuda Toolkit v11.5
- Cudnn Toolkit v8.3.1 for Cuda 11.5

On Jetson Nano

- JetPack 4.6 OS

7.4 Distributing .NET applications

For .NET applications the .NET librairy *id3.Face.dll* shall be distributed in addition to the Windows native libraries.

7.5 Distributing Java applications

For Java applications the java archive file *eu.id3.face.jar* shall be distributed in addition to the native libraries built for the desired platform (Windows or Linux).

7.6 Distributing Android applications

For Android applications this SDK is provided as an android archive file *eu.id3.face.aar*. This archive contains java classes and the native libraries for both armeabi-v7a and arm64_v8a architecture. It must be included along with your application. Please refer to the samples for more information on android usage.

7.7 Runtime Licenses

To deploy a product developed with id3 Face SDK 9.0.4, the integrator should obtain runtime licenses corresponding to the modules and functionalities used by the application. Please see sample for licence management strategies.

8 Technical Support

id3 Technologies provides customer support for a one-year period.

If you have any questions or suggestions regarding this product during this period, you can use the customer web portal on the internet: https://portal.id3.eu

Support email: support-face@id3.eu

id3 Technologies
5, rue de la Verrerie
38120 Le Fontanil-Cornillon
FRANCE

Tel: +33 (0)4 76 75 75 85
Fax: +33 (0)4 76 75 52 30

Internet: http://www.id3.eu
Contact: contact@id3.eu