



Biometric identity
verification for
large-scale
high-security apps

Face Verification SDK



Face Verification SDK

Biometric identity verification for large-scale high-security applications

Document updated on **February 9, 2018**

CONTENTS

Features and capabilities	3
Technical information and specifications	4
System requirements	6
Reliability tests	9
Face Verification Trial SDK and related products	11
Licensing Face Verification SDK	12
Prices	14

The Face Verification SDK is designed for integration of facial authentication into enterprise and consumer applications for mobile devices and PCs. The simple API of the library component helps to implement solutions like payment, e-services and all other apps that need enhanced security through biometric face recognition, while keeping their overall size small for easy deployment to millions of users.

Different liveness detection functionalities are included to implement anti-spoofing mechanism with the possibility of configuring the balance between security and usability of the application.

Available for new and existing customers of VeriLook SDK and MegaMatcher SDK on Android, iOS, Microsoft Windows, Mac OS X and Linux platforms.

- Compact library for deployment on mobile devices.
- Based on VeriLook technology with millions of deployments worldwide.
- Live face detection prevents spoofing.
- Android, iOS, Microsoft Windows, Mac OS X and Linux supported.
- Programming samples in multiple languages included.
- Reasonable prices, flexible licensing and free customer support.



Features and Capabilities

The Face Verification SDK is intended for developing applications which perform end-user identity verification in mass scale systems like:

- online banking and e-shops;
- government e-services;
- social networks and media sharing services.

The Face Verification SDK is based on the VeriLook algorithm, which provides advanced face localization, enrollment and matching using robust digital image processing algorithms based on deep neural networks. The SDK offers these features for large-scale identity verification systems:

- **Live face detection.** A conventional face identification system can be tricked by placing a photo in front of the camera. Face Verification SDK is able to prevent this kind of security breach by determining whether a face in a video stream is “live” or a photograph. The liveness detection can be performed in passive mode, when the engine evaluates certain facial features, and in active mode, when the engine evaluates user’s response to perform actions like blinking or head movements.
- **Face image quality determination.** A quality threshold can be used during face enrollment to ensure that only the best quality face template will be stored into database.
- **Tolerance to face position.** The Face Verification SDK allows head roll, pitch and yaw variation up to 15 degrees in each direction from the frontal position.
- **Multiple samples of the same face.** Biometric template record can contain multiple face samples belonging to the same person. These samples can be enrolled from different sources and at different times, thus allowing improvement in matching quality. For example a person might be enrolled with and without beard or mustache, etc.
- **Features generalization mode.** This mode generates the collection of the generalized face features from several images of the same subject. Then, each face image is processed, features are extracted, and the collections of features are analyzed and combined into a single generalized features collection, which is written to the database. This way, the enrolled feature template is more reliable and the face recognition quality increases considerably.



Technical Information and Specifications

The Face Verification SDK provides certain capabilities for facial recognition applications, including high-level API for all operations and face liveness check. There are also certain requirements for facial image and posture.

General specifications

The following operations are available via the high-level API:

- Face **enrollment** into the internal database – an image with a face is captured from a camera, the face template is extracted from the image and saved into the database.
 - Up to 10 records can be stored in the database.
 - Custom meta-information (like person's name) can be provided during calling this operation to store it in the database together with the face template.
- Face **verification** against a specific face from the database – an image with a face is captured from a camera, the face template is extracted from the image and matched against the template stored in the specified database record.
 - Integrators can enable or disable face liveness detection to prevent fraud attempts with a photo.
 - Parameters like template size or matching quality threshold can be modified before calling the enrollment or verification operations.
- Database record removal

Basic Recommendations for facial image and posture

The face recognition accuracy heavily depends on the quality of a face image.

Image quality during enrollment is important, as it influences the quality of the face template.

- **32 pixels is the recommended minimal distance between eyes** for a face on a video stream to perform face template extraction reliably. **64 pixels or more** recommended for better face recognition results. Note that this distance should be **native**, not achieved by resizing the video frames.
- **Several face enrollments** are recommended for better facial template quality which results in improvement of recognition quality and reliability.
- **Additional enrollments** may be needed when **facial hair** style changes, especially when beard or mustache is grown or shaved off.
- The face recognition engine is intended for usage with near-frontal face images and has certain tolerance to face posture:
 - head **roll** (tilt) – ± 15 degrees;
 - head **pitch** (nod) – ± 15 degrees from frontal position.
 - head **yaw** (bobble) – ± 15 degrees from frontal position.

Continued on the next page



Live Face Detection

A live video stream from a camera is required for face liveness check:

- When the liveness check is enabled, it is performed by the face engine before feature extraction. If the face in the stream **fails** to qualify as “live”, the features are **not extracted**.
- Only **one face should be visible** in these frames.
- Users can enable these liveness check modes:
 - **Active** – the engine requests the user to perform certain actions like blinking or moving one’s head.
 - 5 frames per second or better frame rate required.
 - This mode can work with both colored and grayscale images.
 - This mode requires the user to perform all requested actions to pass the liveness check.
 - **Passive** – the engine analyzes certain facial features while the user stays still in front of the camera for a short period of time.
 - Colored images are required for this mode.
 - 10 frames per second or better frame rate required.
 - Better score is achieved when users do not move at all.
 - **Passive then active** – the engine first tries the passive liveness check, and if it fails, tries the active check. This mode requires colored images.
 - **Simple** – the engine requires user to turn head from side to side while looking at camera.
 - 5 frames per second or better frame rate recommended.
 - This mode can work with both colored and grayscale images.



System requirements

There are specific requirements for each platform which will run applications based on the Face Verification SDK. Note that VeriLook-based applications have different system requirements.

Microsoft Windows platform requirements

- Microsoft Windows **Vista / 7 / 8 / 10**, 32-bit or 64-bit.
- PC or laptop with **x86 (32-bit)** or **x86-64 (64-bit)** compatible processors.
 - 2 GHz or better processor is recommended.
 - **SSE2 support is required.** Processors that do not support SSE2 cannot run the VeriLook algorithm used in the Face Verification SDK. Please check if a particular processor model supports SSE2 instruction set.
- At least **128 MB of free RAM** should be available for the application. Additional RAM is required for applications that perform 1-to-many identification, as all biometric templates need to be stored in RAM for matching. For example, **10,000 templates** (each with 1 face record) require about **50 MB of additional RAM**.
- **Free space on hard disk drive (HDD):**
 - at least 1 GB required for the development.
 - 30 MB required for the component deployment.
- A **camera or webcam** which is accessible using **DirectShow** interface.
- Microsoft **.NET framework 4.5** or newer (for .NET components usage).
- One of following **development environments** for application development:
 - Microsoft Visual Studio 2012 or newer (for application development under C/C++, C#, Visual Basic .Net)
 - Sun Java 1.7 SDK or later.



Android platform requirements

- A smartphone or tablet that is running **Android 4.4 (API level 19)** OS or newer.
 - API level 22 is the recommended **target** for code compilation.
 - If you have a custom Android-based device or development board, contact us to find out if it is supported.
- ARM-based **1.5 GHz processor recommended** for face processing in the specified time. Slower processors may be also used, but the face processing will take longer time.
- At least **20 MB of free RAM** should be available for the application.
- **30 MB of free storage** space (built-in flash or external memory card) required for the component deployment for each separate application.
- Any smartphone's or tablet's **built-in camera** which is supported by Android OS. The camera should have at least 0.3 MegaPixel (640 x 480 pixels) resolution.
- **PC-side development** environment requirements:
 - Java SE JDK 6 (or higher)
 - Eclipse Indigo (3.7) IDE
 - Android development environment (at least API level 19 required)
 - One of the following build automation systems:
 - Apache Maven 3.1.x or newer
 - Gradle 2.10 or newer
 - Internet connection for activating the Face Verification component licenses

iOS platform requirements

- One of the following devices, running **iOS 8.0** or newer:
 - **iPhone 5** or newer iPhone.
 - **iPad 2** or newer iPad, including iPad Mini and iPad Air models.
- At least **20 MB of free RAM** should be available for the application.
- **30 MB of free storage** space (built-in flash or external memory card) required for the component deployment for each separate application.
- **Development environment** requirements:
 - a Mac running Mac OS X 10.10.x or newer.
 - Xcode 6.4 or newer.



Mac OS X platform requirements

- A Mac running **Mac OS X 10.7.x** or newer. 2 GHz or better processor is recommended.
- At least **128 MB of free RAM** should be available for the application.
- **Free space on hard disk drive (HDD):**
 - at least 1 GB required for the development.
 - 30 MB required for the component deployment.
- A **camera or webcam** which is accessible using **GStreamer** interface.
- Specific requirements for **application development:**
 - XCode 4.3 or newer
 - GNU Make 3.81 or newer (to build samples and tutorials development)
 - Sun Java 1.7 SDK or later.

Linux x86 / x86-64 platform requirements

- Linux 2.6 or newer kernel (32-bit or 64-bit) is required. **Linux 3.0 kernel** or newer is recommended.
- PC or laptop with **x86 (32-bit)** or **x86-64 (64-bit)** compatible processors.
 - 2 GHz or better processor is recommended.
 - **SSE2 support is required.** Processors that do not support SSE2 cannot run the VeriLook algorithm used in the Face Verification SDK. Please check if a particular processor model supports SSE2 instruction set.
- At least **128 MB of free RAM** should be available for the application.
- **Free space on hard disk drive (HDD):**
 - at least 1 GB required for the development.
 - 30 MB required for the component deployment.
- A **camera or webcam** which is accessible using **GStreamer** interface.
- glibc 2.11.3 library or newer
- GStreamer 1.2.2 or newer with gst-plugin-base and gst-plugin-good is required for face capture using camera/webcam or rtsp video. GStreamer 1.4.x or newer is recommended.
- Specific requirements for **application development:**
 - GCC-4.4.x or newer
 - GNU Make 3.81 or newer (to build samples and tutorials development)
 - Sun Java 1.7 SDK or later.



Reliability Tests

The Face Verification SDK is **based on the VeriLook face recognition technology**. We present the testing results to show the VeriLook 10.0 algorithm template matching reliability evaluations. The following **public datasets** were used:

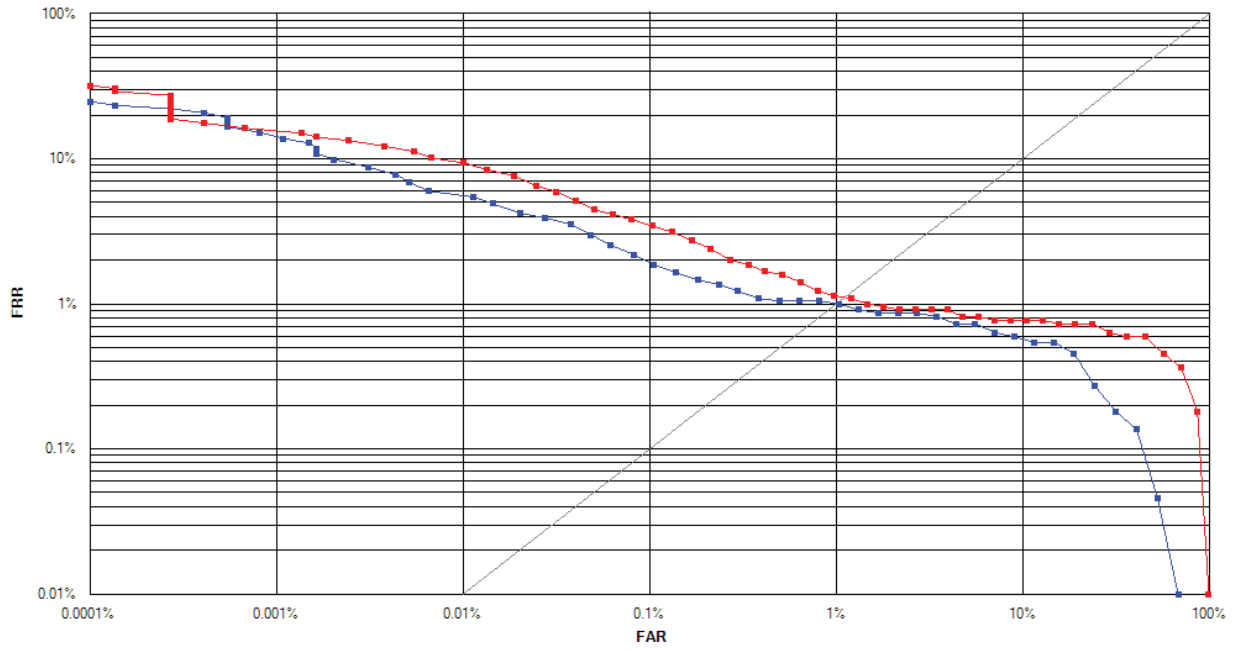
- NIST Special Database 32 - Multiple Encounter Dataset (MEDS-II). - <http://www.nist.gov/itl/iad/ig/sd32.cfm>
 - All full-profile face images from the dataset were removed because they are not supported by VeriLook SDK. This resulted in 1,216 images of 518 persons.
- University of Massachusetts Labeled Faces in the Wild (LFW). - <http://vis-www.cs.umass.edu/lfw/>
 - According to the original protocol, only 6,000 pairs (3,000 genuine and 3,000 impostor) should be used to report the results. But recent algorithms are “very close to the maximum achievable by a perfect classifier” [http://people.cs.umass.edu/~elm/papers/LFW_survey.pdf]. Instead, as Neurotechnology algorithms were not trained on any image from this dataset, verification results on matching each pair of all 13,233 face images of 5,729 persons were chosen to be reported.
 - All identity mistakes, which had been mentioned on the LFW website, were fixed. Also, several not mentioned issues were fixed.
 - Some images from the LFW dataset contained multiple faces. The correct faces for assigned identities were chosen manually to solve these ambiguities.

Two experiments were performed with each dataset:

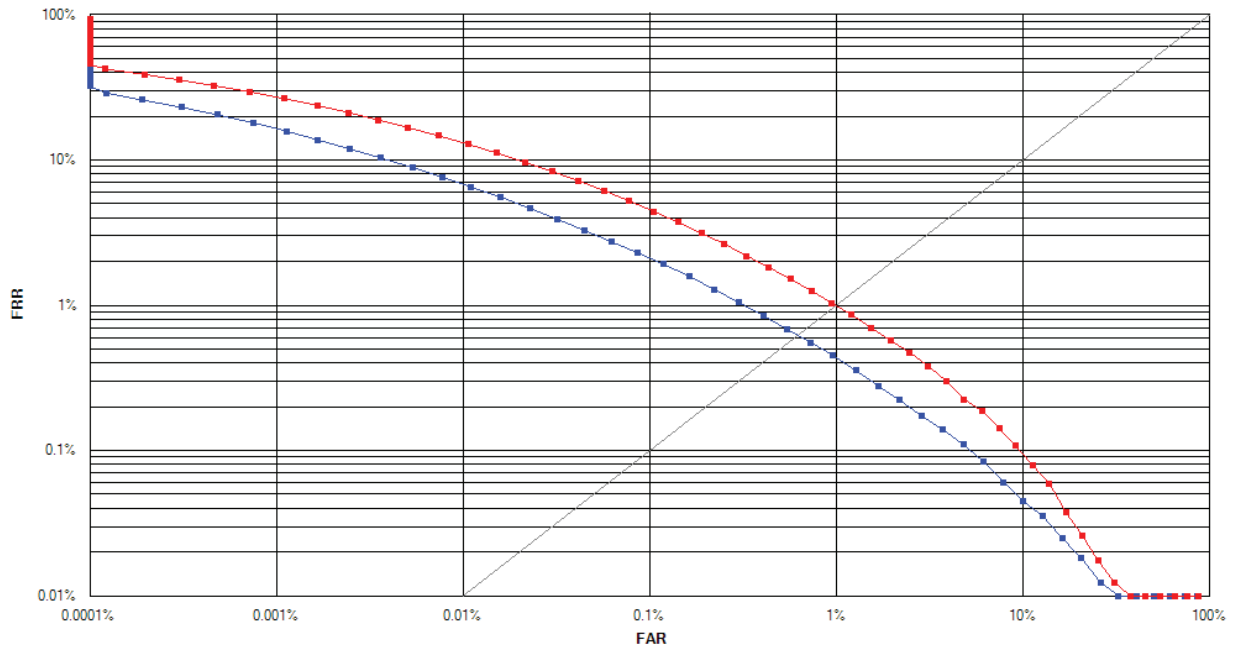
- **Experiment 1** maximized **matching accuracy**. VeriLook 10.0 algorithm reliability in this test is shown on the ROC charts as **blue** curves.
- **Experiment 2** maximized **matching speed**. VeriLook 10.0 algorithm reliability in this test is shown on the ROC charts as **red** curves.

Receiver operation characteristic (**ROC**) curves are usually used to demonstrate the recognition quality of an algorithm. ROC curves show the dependence of false rejection rate (**FRR**) on the false acceptance rate (**FAR**). Equal error rate (**EER**) is the rate at which both FAR and FRR are equal. The ROC charts and the testing results are available on the next pages.

VeriLook 10.0 algorithm testing results with face images from public datasets				
	MEDS-II		LFW	
	Exp. 1	Exp. 2	Exp. 1	Exp. 2
Image count	1216		13233	
Subject count	518		5729	
Session count	1 - 18		1 - 530	
Image size (pixels)	variable		250 x 250	
Template size (bytes)	7128	5066	7128	5066
EER	0.9247 %	1.0550 %	0.6135 %	0.9895 %
FRR at 0.1 % FAR	2.1770 %	3.8100 %	2.2920 %	5.2150 %
FRR at 0.01 % FAR	5.9860 %	10.1100 %	7.5900 %	14.6700 %
FRR at 0.001 % FAR	15.1900 %	16.2400 %	17.9700 %	29.3900 %



VeriLook 10.0 SDK matching engine with face templates from NIST Multiple Encounter Dataset (MEDS-II):
■ Experiment 1, maximized matching accuracy scenario
■ Experiment 2, maximized matching speed scenario



VeriLook 10.0 SDK matching engine with face templates from the University of Massachusetts Labeled Faces in the Wild (LFW) dataset:
■ Experiment 1, maximized matching accuracy scenario
■ Experiment 2, maximized matching speed scenario



Face Verification Trial SDK and Related Products

The Face Verification **30-day SDK Trial** is available for downloading at www.neurotechnology.com/download.html.

These products are related to the Face Verification SDK:

- **VeriLook SDK** – for developing systems that perform **facial identification**. The Face Verification SDK is based on the VeriLook SDK technology and is a part of the VeriLook SDK. VeriLook SDK supports the same platforms as the Face Verification SDK.
- **MegaMatcher SDK** – for development of AFIS or multi-biometric face, fingerprint, iris, voiceprint and palm print identification products.
- **MegaMatcher On Card SDK** – a product for fingerprint, iris and face matching on smart cards.
- **Free Fingerprint Verification SDK** – a freeware SDK intended for adding fingerprint verification functionality into various applications.



Licensing Face Verification SDK

Product Development

An integrator should obtain one of these SDKs to develop an end-user product based on the Face Verification SDK:

- VeriLook 10.0 Standard SDK (EUR 339);
- VeriLook 10.0 Extended SDK (EUR 839);
- MegaMatcher 10.0 Standard SDK (EUR 2590);
- MegaMatcher 10.0 Extended SDK (EUR 4990).

One of the mentioned SDKs needs to be purchased just once and may be used for all projects and by all the developers within the integrator's company.

The Face Verification SDK includes the Face Verification component. One free license for the Face Verification component is included with the Face Verification SDK. Integrators can obtain additional component licenses if more component licenses are required for the development process.

Product Deployment

To deploy their developed products, an integrator needs to obtain licenses for every **computer or device**, where the Face Verification component will be installed together with the integrator's product. Integrators can purchase additional Face Verification component licenses if required at any time.

License activation options

The components are copy-protected. The following license activation options are available:

- **Serial numbers** are used to activate licenses for the Face Verification component on a particular computer or device. The activation is done via the Internet or by email. After activation, the network connection is not required for single computer license usage.

Notes:

 1. Activation by serial number is **not suitable for iOS**.
 2. Activation by serial number is **not suitable for virtual environments**.
- **Internet activation.** A special **license file** is stored on a computer or a mobile or embedded device; the license file allows to run the Face Verification component on that computer or device after **checking** the license over the Internet. **Internet connection** should be available periodically for a short amount of time. A single computer license can be **transferred** to another computer or device by moving the license file there and waiting until the previous activation expires.
- **Volume License Manager.** Licenses may be stored in a volume license manager **dongle**. License activation using volume license manager may be performed without connection to the Internet and is suitable for virtual environments. Volume license manager is **used on site by integrators or end users** to manage licenses for the Face Verification component in the following ways:
 1. **Activating single computer licenses** – An installation license for the Face Verification component will be activated for use on a particular computer. The number of available licenses in the license manager will be decreased by the number of activated licenses.
 2. **Managing single computer licenses via a LAN or the Internet** – The license manager allows the management of installation licenses for the Face Verification component across multiple computers or mobile/embedded devices in a LAN or over the Internet. The number of managed licenses is limited by the number of licenses in the license manager. No license activation is required and the license quantity is not decreased. Once issued, the license is assigned to a specific computer or device on the network.
 3. **Using license manager as a dongle** – A volume license manager containing at least one license for the Face Verification component may be used as a dongle, allowing the component to run on the particular computer where the dongle is attached.



Licenses Validity

All SDK and component licenses are perpetual and do not have expiration. There are no annual fee or any other fees except license purchasing fee. It is possible to move licenses from one computer or device to another. Neurotechnology provides a way to renew the license if the computer undergoes changes due to technical maintenance.

Licensing Agreement

The Licensing Agreement (http://neurotechnology.com/mm_100_sla.html) contains all licensing terms and conditions.

Note that you unambiguously accept this agreement by placing an order using Neurotechnology online ordering service or by email or other means of communications. Please read the agreement before making an order.

Other licensing options

- **VAR License.** The above described licensing model is intended for **end-user** product developers. Integrators who want to develop and sell a development tool based on the Face Verification SDK (with API, programming possibilities, programming samples, etc.), must obtain permission from Neurotechnology and **sign** a special VAR agreement. For more information please contact us.
- **Enterprise License.** The Face Verification SDK enterprise license allows an **unlimited use** of the Face Verification component in end-user products for a specific territory, market segment or project. Specific restrictions would be included in the licensing agreement. The enterprise license price depends on the application size and the number of potential users of the application within the designated territory, market segment or project. For more information please contact us.



Prices

- The prices are **effective June 21, 2017**. The prices may change in the future, so please **download and review the latest version** of the brochure before making an order.
- The Face Verification component licenses are available only for the customers of VeriLook SDK or MegaMatcher SDK.
- Quantity discounts do not accumulate over time.
- Prices do not include local import duties or taxes.
- Product shipping costs depend on delivery country
- Customers with Solution Partner status are eligible for product discounts.

Component prices

Quantity	Price per license
1 - 9	€ 1.75
10 - 19	€ 1.27
20 - 49	€ 1.13
50 - 99	€ 1.00
100 - 199	€ 0.89
200 - 499	€ 0.79
500 - 999	€ 0.69
1000 - 1999	€ 0.62
2000 - 3999	€ 0.56
4000 - 7999	€ 0.50
8000 and more	Please contact us for more information

License management

Volume license manager	€ 16.00
------------------------	---------

Face Verification SDK enterprise license

Face Verification SDK enterprise license	Please contact us for more information
--	--

The Face Verification SDK and related products can be ordered:

- online, at www.neurotechnology.com/cgi-bin/order.cgi
- via a local Neurotechnology distributor; the list of distributors is available at www.neurotechnology.com/distributors.html