Iris identification for stand-alone or Web solutions

VeriEye SDK
VeriEye is available as a software development kit that allows development of stand-alone and Web-based solutions on Microsoft Windows, Linux, macOS, iOS and Android platforms.

- Rapid and accurate iris identification, proven by NIST IREX.
- Robust recognition, even with gazing-away eyes or narrowed eyelids.
- Original proprietary algorithm solves the limitations and drawbacks of existing state-of-the-art algorithms.
- Contact lens detection can prevent spoof with fake iris images.
- Available as multiplatform SDK that supports multiple programming languages.
- Reasonable prices, flexible licensing and free customer support.
Features and Capabilities

Neurotechnology began research and development in the field of eye iris biometrics in 1994. In 2008, Neurotechnology released VeriEye iris recognition algorithm. The next year VeriEye was recognized by NIST as one of the most reliably accurate iris recognition algorithms.

The proprietary algorithm implements advanced iris segmentation, enrollment and matching using robust digital image processing algorithms:

- **Robust iris detection.** Irises are detected even when there are obstructions to the image, visual noise and/or different levels of illumination. Lighting reflections, eyelids and eyelashes obstructions are eliminated. Images with narrowed eyelids or eyes that are gazing away are also accepted.

- **Automatic interlacing detection and correction** results in maximum quality of iris features templates from moving iris images.

- **Gazing-away eyes** are correctly detected on images, segmented and transformed as if it were looking directly into the camera (see Figure 1).

- **Correct iris segmentation** is obtained even under these conditions:
  - **Perfect circles fail.** VeriEye uses active shape models that more precisely model the contours of the eye, as iris boundaries are not modeled by perfect circles.
  - **The centers of the iris inner and outer boundaries are different** (see Figure 2). The iris inner boundary and its center are marked in red, the iris outer boundary and its center are marked in green.
  - **Iris boundaries are definitely not circles and even not ellipses** (see Figure 3) and especially in gazing-away iris images.
  - **Iris boundaries seem to be perfect circles.** The recognition quality can still be improved if boundaries are found more precisely (see Figure 4). Note these slight imperfections when compared to perfect circular white contours.
  - **Iris is partially occluded by eyelids.** The upper and lower lids are marked in red and green correspondingly (see Figure 5).

- **Iris image quality determination and spoof prevention.** The image quality estimation can be used during iris enrollment to ensure that only the best quality iris template will be stored into database. Also, cosmetic (decorative) **contact lens**, which obscure an iris with some artistic or **fake iris** texture and/or change iris color, can be detected.

- **Reliability.** VeriEye algorithm has shown excellent recognition accuracy during the NIST IREX evaluations, as well as during testing on publicly available datasets. (see **Reliability and Performance Tests Results** section).

All presented iris images are taken from CASIA Iris Image Database V2.0 and CASIA Iris Image Database V3.0 collected by the Chinese Academy of Sciences Institute of Automation (CASIA) (www.cbsr.ia.ac.cn/english/IrisDatabases.asp).
**Technology Awards**

VeriEye iris identification technology performance and reliability was proven in several NIST-organized competitions and technology evaluations.

**IREX evaluations by NIST**

- **IREX IX** – in 2018 Neurotechnology’s iris recognition algorithm has been judged by the NIST as the second most accurate among the participants. The accelerated version of the algorithm was nearly 50 times faster than any other matcher in the NIST IREX IX evaluation.

- **IREX IV** – in 2013 Neurotechnology’s iris recognition algorithm has been judged by the NIST as one of the fastest and most accurate among the participants.

- **IREX III** – in 2012 VeriEye iris matching algorithm was the second fastest and provided 3 times higher recognition accuracy than the only faster contender.

- **IREX** – in 2009 VeriEye iris matching algorithm was recognized as the fastest overall with top 3 accuracy rating and up to 7.5 times smaller template than the corresponding contenders.
VeriEye SDK is based on VeriEye iris recognition technology and is intended for biometric systems developers and integrators. The SDK allows rapid development of biometric applications using functionality from the VeriEye algorithm that ensures reliable fast iris identification. VeriEye can be easily integrated into the customer’s security system. The integrator has complete control over SDK data input and output.

VeriEye SDK includes the Device Manager library for working with the supported iris cameras. Integrators can also write plug-ins to support their iris cameras or other devices using the plug-in framework provided with the Device Manager.

VeriEye is available as the following SDKs:

- **VeriEye 11.2 Standard SDK** is designed for PC-based, embedded or mobile biometric application development. It includes Iris Matcher and Extractor component licenses, programming samples and tutorials, iris scanner support modules and software documentation. The SDK allows the development of biometric applications for Microsoft Windows, Linux, macOS, iOS or Android operating systems.

- **VeriEye 11.2 Extended SDK** is designed for biometric web-based and network application development. It includes all features and components of the Standard SDK. Additionally, the SDK contains Iris Client component licenses for PCs and mobile / embedded devices, sample client applications, tutorials and a ready-to-use matching server component.

The table below compares VeriEye Standard SDK and VeriEye Extended SDK. See the licensing model for more information on specific license types.

<table>
<thead>
<tr>
<th>Component licenses included with a specific SDK</th>
<th>VeriEye 11.2 Standard SDK</th>
<th>VeriEye 11.2 Extended SDK</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Iris Matcher</td>
<td>1 single computer license</td>
<td>1 single computer license</td>
</tr>
<tr>
<td>• Mobile Iris Matcher</td>
<td>1 single computer license</td>
<td>1 single computer license</td>
</tr>
<tr>
<td>• Iris Client</td>
<td>3 single computer licenses</td>
<td></td>
</tr>
<tr>
<td>• Mobile Iris Client</td>
<td>3 single computer licenses</td>
<td></td>
</tr>
<tr>
<td>• Iris Extractor</td>
<td>1 single computer license</td>
<td>1 single computer license</td>
</tr>
<tr>
<td>• Mobile Iris Extractor</td>
<td>1 single computer license</td>
<td>1 single computer license</td>
</tr>
<tr>
<td>• Matching Server</td>
<td></td>
<td>+</td>
</tr>
</tbody>
</table>

VeriEye 11.2 SDK includes programming samples and tutorials that show how to use the components of the SDK to perform face template extraction or matching against other templates. The samples and tutorials are available for these programming languages and platforms:

<table>
<thead>
<tr>
<th>Programming samples and tutorials</th>
<th>Windows</th>
<th>Linux</th>
<th>macOS</th>
<th>iOS</th>
<th>Android</th>
</tr>
</thead>
<tbody>
<tr>
<td>• C/C++</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>• Objective-C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>+</td>
</tr>
<tr>
<td>• C#</td>
<td>+</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Java</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>• Visual Basic .NET</td>
<td>+</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Biometric Components Description

Iris Matcher

The Iris Matcher performs iris template matching in 1-to-1 (verification) and 1-to-many (identification) modes. Also the Iris Matcher component includes fused matching algorithm that allows to increase template matching reliability by:

- matching templates that contain 2 iris records;
- matching templates that contain fingerprint, face, voiceprint and/or iris records (note that matching fingerprints, faces and voiceprints requires Fingerprint Matcher, Face Matcher and Voice Matcher components correspondingly);

The Iris Matcher component matches **40,000 irises per second**.

One Iris Matcher license is included with VeriEye 11.2 Standard SDK and VeriEye 11.2 Extended SDK. The license can be used on Microsoft Windows, Linux x86/x86_64 or macOS platform. More licenses for this component can be purchased any time by VeriEye 11.2 SDK customers.

Mobile Iris Matcher

The Mobile Iris Matcher performs iris template matching in 1-to-1 (verification) and 1-to-many (identification) modes. It matches **3,000 irises per second** and is designed to be used in embedded or mobile biometric systems, which run on Android or iOS or ARM Linux devices. The Android devices should be based on at least Snapdragon S4 system-on-chip (Krait 300 processor with 4 cores running at 1.51 GHz).

One Mobile Iris Matcher license is included with VeriEye 11.2 Standard SDK and VeriEye 11.2 Extended SDK. The license can be used on Android, iOS or ARM Linux platform. More licenses for this component can be purchased any time by VeriEye 11.2 SDK customers.

Iris Extractor

Iris Extractor creates iris templates from eye images.

The component extracts a single iris template in **1.34 seconds**. The specified performance requires a PC or laptop with at least Intel Core i7-4771 processor.

One Iris Extractor license is included with VeriEye 11.2 Standard SDK and VeriEye 11.2 Extended SDK. The license can be used on Microsoft Windows, Linux x86/x86_64 or macOS platform. More licenses for this component can be purchased any time by VeriEye 11.2 SDK customers.

Mobile Iris Extractor

The Mobile Iris Extractor component has the same functionality as the Iris Extractor and is designed to run on Android or iOS or ARM Linux devices. The Android devices should be based on at least Snapdragon S4 system-on-chip (Krait 300 processor with 4 cores running at 1.51 GHz). The component extracts a single iris template in **1.34 seconds**.

One Mobile Iris Extractor license is included with VeriEye 11.2 Standard SDK and VeriEye 11.2 Extended SDK. The license can be used on Android, iOS or ARM Linux platform. More licenses for this component can be purchased any time by VeriEye 11.2 SDK customers.
Iris Client

The Iris Client component includes the capabilities of Iris Extractor component for iris templates creation from eye images, as well as image formats support based on biometric standards.

The Iris Client extracts a single iris template in **0.6 seconds**. The specified performance requires a PC or laptop with at least Intel Core i7-4771 processor.

The component also allows to integrate JPEG 2000 image format support into applications based on VeriEye SDK.

The Iris Client component allows to integrate support for iris image format standards and additional image formats with new or existing biometric systems based on VeriEye SDK.

These biometric standards are supported:

- BioAPI 2.0 (ISO/IEC 19784-1:2006) (Framework and Biometric Service Provider for iris identification engine)
- CBEFF V1.2 (ANSI INCITS 398-2008) (Common Biometric Exchange Formats Framework)
- ISO/IEC 19794-6:2005 (Biometric Data Interchange Formats - Iris Image Data)
- ISO/IEC 19794-6:2011 with Cor. 1:2012
- ISO/IEC 29794-6:2015 (Biometric Sample Quality - Iris Image Data)
- ANSI/INCITS 379-2004 (Iris Image Interchange Format)
- ANSI/NIST-ITL 1-2007 (Data Format for the Interchange of Fingerprint, Facial, & Other Biometric Information)
- ANSI/NIST-ITL 1a-2009 (Data Format for the Interchange of Fingerprint, Facial, & Other Biometric Information)
- ANSI/NIST-ITL 1-2011 (Data Format for the Interchange of Fingerprint, Facial, & Other Biometric Information)
- ANSI/NIST-ITL 1-2011 Update:2015 (Data Format for the Interchange of Fingerprint, Facial, & Other Biometric Information)

All functionalities of the Iris Client component can be used from C/C++, C# and Java applications on all supported platforms. .NET wrappers of Windows libraries are provided for .NET developers.

Three licenses for the Iris Client component are included with VeriEye 11.2 Extended SDK. The licenses can be used on Microsoft Windows, Linux x86/x86_64 or macOS platform. More licenses for this component can be purchased any time by VeriEye 11.2 Extended SDK customers.

Mobile Iris Client

The Mobile Iris Client component has the same functionality as the Iris Client and is designed to run on Android or iOS or ARM Linux devices. The Android devices should be based on at least Snapdragon S4 system-on-chip (Krait 300 processor with 4 cores running at 1.51 GHz). The component extracts a single iris template in **1.2 seconds**.

Three licenses for the Mobile Iris Client component are included with VeriEye 11.2 Extended SDK. The licenses can be used on Android, iOS or ARM Linux platform. More licenses for this component can be purchased any time by VeriEye 11.2 Extended SDK customers.
Matching Server

The Matching Server is ready-to-use software intended for building moderate size web-based and other network-based systems like local single- or multi-biometric identification system. The Server software runs on a server PC and allows to perform the biometric template matching on server side using Iris Matcher component.

Multi-biometric matching can be enabled by running components for iris, fingerprint, face and voiceprint matching on the same machine.

Client communication module that allows sending a task to the Matching Server, querying status of the task, getting the results and removing the task from server, is included with MegaMatcher 11.2 SDK, VeriFinger 11.2 SDK, VeriLook 11.2 SDK, VeriSpeak 11.2 SDK and VeriEye 11.2 SDK. This module hides all low level communications and provides high-level API for the developer.

The components and database support modules with source codes included for Matching Server component are listed in the table below. Custom modules for working with other databases can also be developed by integrator and used with the Matching Server software.

The table below shows what components are available with Matching Server software.

<table>
<thead>
<tr>
<th>Components</th>
<th>Microsoft Windows</th>
<th>Linux</th>
<th>macOS</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Matching server software</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>• Server administration tool API</td>
<td>+</td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>• Source code of sample web server software</td>
<td>+</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Database support modules</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Microsoft SQL Server</td>
<td>+</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PostgreSQL</td>
<td>+</td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>• MySQL</td>
<td>+</td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>• Oracle</td>
<td>+</td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>• SQLite</td>
<td>+</td>
<td></td>
<td>+</td>
</tr>
<tr>
<td>Programming samples</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• C# client</td>
<td>+</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Visual Basic .NET client</td>
<td>+</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Java web client</td>
<td>+</td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>Programming tutorials</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• C/C++</td>
<td>+</td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>• C#</td>
<td>+</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Visual Basic .NET</td>
<td>+</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The Matching Server component requires a special license that allows to run the component on all machines that run the fingerprint, face, iris, voiceprint or palm print matching components obtained by an integrator. The Matching Server software is included with VeriEye 11.2 Extended SDK.

Also the Matching Server component is included with these Neurotechnology SDKs (see their brochures for more info):

- MegaMatcher 11.2 Standard or MegaMatcher 11.2 Extended SDK;
- VeriFinger 11.2 Extended SDK;
- VeriLook 11.2 Extended SDK.
- VeriSpeak 11.2 Extended SDK.
## Supported Iris Capture Cameras

The table below explains which eye iris scanners are supported by VeriEye SDK under different operating systems.

We are always looking for scanners’ manufacturers to include the support for their iris scanners to our products. Please, contact us for more details.

Integrators or scanner manufacturers can also write plug-ins for the Device Manager from the VeriEye SDK to support their iris cameras using the provided plug-in framework. The SDK documentation contains more information about the plug-in framework.

<table>
<thead>
<tr>
<th>Iris capture cameras</th>
<th>Microsoft Windows 7 / 8 / 10</th>
<th>Linux</th>
<th>Android</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>32 bit</td>
<td>64 bit</td>
<td>32 bit</td>
</tr>
<tr>
<td>CMITech BMT-20 / EMX-30</td>
<td>+</td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>Credence ID Trident</td>
<td></td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>Cross Match I Scan 2</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Iris ID iCAM T10 / iCAM TD100</td>
<td>+</td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>Intech InShield USB MK 2120U / InShield-USB BK 2121U</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Intech InMagic1000BK</td>
<td>+</td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>Mantra MIS100V2</td>
<td>+</td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>VistaFA2 / VistaFA2E / VistaEY2 / VistaEY2-02 iris &amp; face cameras</td>
<td>+</td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>VistaEY2H iris camera</td>
<td>+</td>
<td>+</td>
<td></td>
</tr>
</tbody>
</table>

(1) Can be used on 64-bit OS, but only in 32-bit applications.
(2) The device has pre-installed Android OS.
System requirements

There are specific requirements for each platform which will run VeriEye-based applications.

Microsoft Windows platform requirements

- PC or laptop with x86-64 (64-bit) compatible processors.
  - 2 GHz or better processor is recommended.
  - x86 (32-bit) processors can still be used, but the algorithm will not provide the specified performance.
  - **AVX2 support is highly recommended.** Processors that do not support AVX2 will still run the VeriEye algorithms, but in a mode, which will not provide the specified performance. Most modern processors support this instruction set, but please check if a particular processor model supports it.
- At least **512 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.
- **Iris scanner:**
  - VeriEye SDK includes support modules for several iris scanners under Microsoft Windows platform. See the previous chapter for details.
  - Iris images in BMP, JPG or PNG formats can be processed thus almost any third-party iris capturing hardware can be used with the VeriEye technology if it generates images in the mentioned formats.
  - Integrators may also write plug-ins to support their iris cameras using the plug-in framework provided with the Device Manager from the VeriEye SDK. The integrators should note, that the most accurate iris recognition is achievable only when iris images are captured with near-infrared cameras and appropriate illumination. However, it is still possible to recognize irises with reasonable accuracy, when the irises are captured with regular cameras, using proper illumination and focus, and choosing proper environment. See our testing results for details.
- **Database engine** or connection with it. VeriEye templates can be saved into any DB (including files) supporting binary data saving. VeriEye Extended SDK contains the following support modules for Matching Server on Microsoft Windows platform: **Microsoft SQL Server, MySQL, Oracle, PostgreSQL** and **SQLite**.
- **Network/LAN connection (TCP/IP)** for client/server applications. Also, network connection is required for using Matching server component (included in VeriEye Extended SDK). VeriEye SDK does not provide communication encryption with the Matching server, therefore, integrators should secure the communication by themselves.
- 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.

Database engine or connection with it. VeriEye templates can be saved into any DB (including files) supporting binary data saving. VeriEye Extended SDK contains the following support modules for Matching Server on Microsoft Windows platform: **Microsoft SQL Server, MySQL, Oracle, PostgreSQL** and **SQLite**.

Network/LAN connection (TCP/IP) for client/server applications. Also, network connection is required for using Matching server component (included in VeriEye Extended SDK). VeriEye SDK does not provide communication encryption with the Matching server, therefore, integrators should secure the communication by themselves.

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.
  - 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 iris processing in the specified time. Slower processors may also be used, but the iris processing will take longer time.
- At least **256 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.
- **Iris scanner:**
  - VeriEye SDK includes support modules for several iris scanners under Android platform. See the previous chapter for details.
  - Iris images in **BMP, JPG** or **PNG** formats can be processed thus almost any third-party iris capturing hardware can be used with the VeriEye technology if it generates images in the mentioned formats.
  - Integrators may also write **plug-ins to support their iris cameras** using the plug-in framework provided with the Device Manager from the VeriEye SDK. The integrators should note, that the most accurate iris recognition is achievable only when iris images are captured with near-infrared cameras and appropriate illumination. However, it is still possible to recognize irises with reasonable accuracy, when the irises are captured with regular cameras, using proper illumination and focus, and choosing proper environment. See our testing results for details.
- **Network/LAN connection (TCP/IP)** for client/server applications. Also, network connection is required for using Matching server component (included in VeriEye Extended SDK). VeriEye SDK does not provide communication encryption with the Matching server, therefore, integrators should secure the communication by themselves.
- **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)
  - Gradle 4.6 build automation system or newer
  - Internet connection for activating VeriEye component licenses
**iOS platform requirements**

- One of the following devices, running **iOS 11.0** or newer:
  - **iPhone 5S** or newer iPhone.
  - **iPad Air** or newer iPad models.

- At least **256 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.

- **Iris scanner.**
  - At the moment iris scanner support on iOS platform should be implemented by integrators. The integrators should note, that the most accurate iris recognition is achievable only when iris images are captured with near-infrared cameras and appropriate illumination. However, it is still possible to recognize irises with reasonable accuracy, when the irises are captured with cameras, which are built in smartphones or tablets, using proper illumination and focus, and choosing proper environment. See our testing results for details.
  - Iris images in **BMP, JPG** or **PNG** formats can be processed thus almost any third-party iris capturing hardware can be used with the VeriEye technology if it generates images in the mentioned formats.

- **Network/LAN connection (TCP/IP)** for client/server applications. Also, network connection is required for using Matching server component (included in VeriEye Extended SDK). VeriEye SDK does not provide communication encryption with the Matching server, therefore, integrators should secure the communication by themselves.

- **Development environment** requirements:
  - a Mac running macOS 10.12.6 or newer.
  - Xcode 9.x or newer.
macOS platform requirements

- A Mac running macOS 10.12.6 or newer.
  - 2 GHz or better processor is recommended.
  - AVX2 support is highly recommended. Processors that do not support AVX2 will still run the VeriEye algorithms, but in a mode, which will not provide the specified performance. Most modern processors support this instruction set, but please check if a particular processor model supports it.

- At least 512 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.

- Iris scanner.
  - At the moment iris scanner support on Mac OS X platform should be implemented by integrators. The integrators should note, that the most accurate iris recognition is achievable only when iris images are captured with near-infrared cameras and appropriate illumination. However, it is still possible to recognize irises with reasonable accuracy, when the irises are captured with cameras, which are built in smartphones or tablets, using proper illumination and focus, and choosing proper environment. See our testing results for details.
  - Iris images in BMP, JPG or PNG formats can be processed thus almost any third-party iris capturing hardware can be used with the VeriEye technology if it generates images in the mentioned formats.

- Database engine or connection with it. VeriEye templates can be saved into any DB (including files) supporting binary data saving. VeriEye Extended SDK contains SQLite support modules for Matching Server on macOS platform.

- Network/LAN connection (TCP/IP) for client/server applications. Also, network connection is required for using Matching server component (included in VeriEye Extended SDK). VeriEye SDK does not provide communication encryption with the Matching server, therefore, integrators should secure the communication by themselves.

- Specific requirements for application development:
  - XCode 6.x or newer
  - GNU Make 3.81 or newer (to build samples and tutorials development)
  - Sun Java 1.8 SDK or later.
Linux x86 / x86-64 platform requirements

- **Linux 3.10 kernel** or newer is required.
- PC or laptop with **x86-64 (64-bit)** compatible processors.
  - 2 GHz or better processor is recommended.
  - x86 (32-bit) processors can still be used, but the algorithm will not provide the specified performance.
  - **AVX2 support is highly recommended.** Processors that do not support AVX2 will still run the VeriEye algorithms, but in a mode, which will not provide the specified performance. Most modern processors support this instruction set, but please check if a particular processor model supports it.
- At least **512 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.
- **Iris scanner.**
  - VeriEye SDK includes support modules for several iris scanners under Linux platform.
  - Iris images in **BMP, JPG or PNG** formats can be processed thus almost any third-party iris capturing hardware can be used with the VeriEye technology if it generates images in the mentioned formats.
  - Integrators may also write **plug-ins to support their iris cameras** using the plug-in framework provided with the Device Manager from the VeriEye SDK. The integrators should note, that the most accurate iris recognition is achievable only when iris images are captured with near-infrared cameras and appropriate illumination. However, it is still possible to recognize irises with reasonable accuracy, when the irises are captured with cameras, which are built in smartphones or tablets, using proper illumination and focus, and choosing proper environment. See our testing results for details.
  - **glibc 2.17 library or newer**
- **Database engine** or connection with it. VeriEye templates can be saved into any DB (including files) supporting binary data saving. VeriEye Extended SDK contains **MySQL, Oracle, PostgreSQL** and **SQLite** support modules for Matching Server on Linux x86 / x86-64 platforms.
- **Network/LAN connection (TCP/IP)** for client/server applications. Also, network connection is required for using Matching server component (included in VeriEye Extended SDK). VeriEye SDK does not provide communication encryption with the Matching server, therefore, integrators should secure the communication by themselves.
- **Specific requirements for application development:**
  - gcc 4.8 or newer
  - GNU Make 3.81 or newer (to build samples and tutorials development)
  - Sun Java 1.8 SDK or later.
ARM Linux platform requirements

We recommend to contact us and report the specifications of a target device to find out if it will be suitable for running VeriEye-based applications.

There is a list of common requirements for ARM Linux platform:

- A device with ARM-based processor, running Linux 3.2 kernel or newer.
- ARM-based 1.5 GHz processor recommended for iris processing in the specified time.
  - ARMHF architecture (EABI 32-bit hard-float ARMv7) is required.
  - Lower clock-rate processors may be also used, but the iris processing will take longer time.
- At least 256 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.
- Iris scanner.
  - At the moment iris scanner support on ARM Linux platform should be implemented by integrators. The integrators should note, that the most accurate iris recognition is achievable only when iris images are captured with near-infrared cameras and appropriate illumination. However, it is still possible to recognize irises with reasonable accuracy, when the irises are captured with cameras, which are built in smartphones or tablets, using proper illumination and focus, and choosing proper environment. See our testing results for details.
  - Iris images in BMP, JPG or PNG formats can be processed thus almost any third-party iris capturing hardware can be used with the VeriEye technology if it generates images in the mentioned formats.
- glibc 2.17 library or newer
- Network/LAN connection (TCP/IP) for client/server applications. Also, network connection is required for using Matching server component (included in VeriEye Extended SDK). VeriEye SDK does not provide communication encryption with the Matching server, therefore, integrators should secure the communication by themselves.
- Development environment requirements:
  - gcc 4.8 or newer
  - GNU Make 3.81 or newer (to build samples and tutorials development)
  - Sun Java 1.8 SDK or later.
## Technical Specifications

*64 pixels* is the minimal radius of circle containing full iris texture, that is required for iris template extraction.

**Near-infrared** spectral region is recommended for iris image capture.

All iris templates should be loaded into RAM before identification, thus the maximum iris template database size is limited by the amount of available RAM.

VeriEye biometric template extraction and matching algorithm is designed to run on **multi-core processors** allowing to reach maximum possible performance on the used hardware.

### VeriEye 11.2 iris engine specifications

<table>
<thead>
<tr>
<th>Template extraction components</th>
<th>Embedded / mobile platform (1)</th>
<th>PC-based platform (2)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mobile Iris Extractor</td>
<td>Mobile Iris Client</td>
</tr>
<tr>
<td>Iris template extraction time (seconds)</td>
<td>1.34</td>
<td>1.20</td>
</tr>
<tr>
<td>Template matching components</td>
<td>Mobile Iris Matcher</td>
<td>Iris Matcher</td>
</tr>
<tr>
<td>Matching speed (Irises per second)</td>
<td>3,000</td>
<td></td>
</tr>
<tr>
<td>Single iris record size in a template (bytes)</td>
<td></td>
<td>2,348</td>
</tr>
</tbody>
</table>

**Notes:**

1. Requires to be run on iOS or Android devices based on at least Snapdragon S4 system-on-chip with Krait 300 processor (1.51 GHz).
2. Requires to be run on PC or laptop with at least Intel i7-4771 processor (2.67 GHz).
Reliability Test Results

We present the testing results to show VeriEye 11.2 template matching algorithm reliability evaluations. Iris images from several standard datasets were used for testing, thus the testing results can be compared with testing results of other algorithms. All datasets contained iris images with 640 x 480 pixels size.

### Iris image datasets used for VeriEye 11.2 algorithm testing

<table>
<thead>
<tr>
<th>Dataset</th>
<th>ICE2005 Exp1</th>
<th>ND-IRIS-0405</th>
<th>IRISDB1600</th>
<th>MobileirisV10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image count</td>
<td>1,425</td>
<td>64,980</td>
<td>24,361</td>
<td>3,290</td>
</tr>
<tr>
<td>Subject count</td>
<td>124</td>
<td>356</td>
<td>624</td>
<td>70</td>
</tr>
<tr>
<td>Unique iris count</td>
<td>124</td>
<td>712</td>
<td>1231</td>
<td>135</td>
</tr>
<tr>
<td>Session count</td>
<td>1 - 31</td>
<td>4 - 291</td>
<td>1 - 40</td>
<td>6 - 42</td>
</tr>
</tbody>
</table>

**Notes:**

1. The ICE2005 dataset was collected by the National Institute of Standards and Technology (NIST). Near-infrared spectrum equipment was used for iris capture. ICE2005 Exp1 is a subset, which contains right iris images.
2. The ND-IRIS-0405 was collected by the University of Notre Dame. Near-infrared spectrum equipment was used for iris capture.
3. The IRISDB1600 was collected by the University of Bath. Near-infrared spectrum equipment was used for iris capture. The full IRISDB1600 dataset contains 31,997 images (image size 1280x960 pixels), which represented 799 unique persons and 1,598 unique irises. A subset used in this test was preprocessed similar to NIST IREX experiments – the images were downsampled to 640x480 via 2x2 neighborhood averaging, and all images containing irises with diameters larger than 340 pixels were removed.
4. The MobileirisV10 dataset was collected by the Warsaw University of Technology. The iris image collection was performed using regular, visible light spectrum camera built-in into Apple iPhone 5S smartphone. Colored images were collected with the camera. The images were resized to 640x480 pixels and converted to grayscale. See the scientific paper ([https://zbum.ia.pw.edu.pl/PAPERS/Wilga2015_SmartphoneIris.pdf](https://zbum.ia.pw.edu.pl/PAPERS/Wilga2015_SmartphoneIris.pdf)) for more details.

Two tests were performed with each database:

- **Test 1** maximized matching accuracy. VeriEye 11.2 algorithm reliability in this test is shown as blue curves on the ROC charts.
- **Test 2** maximized matching speed. VeriEye 11.2 algorithm reliability in this test is shown as red curves on the ROC charts.

The iris rotation tolerance was set to ±15° in all tests.

### VeriEye 11.2 algorithm reliability testing results

<table>
<thead>
<tr>
<th>Dataset</th>
<th>ICE2005 Exp1</th>
<th>ND-IRIS-0405</th>
<th>IRISDB1600</th>
<th>MobileirisV10</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Test 1</td>
<td>Test 2</td>
<td>Test 1</td>
<td>Test 2</td>
</tr>
<tr>
<td>FRR at 0.01 % FAR</td>
<td>0.0082 %</td>
<td>0.0819 %</td>
<td>1.0410 %</td>
<td>1.2620 %</td>
</tr>
<tr>
<td>FRR at 0.001 % FAR</td>
<td>0.0246 %</td>
<td>0.0983 %</td>
<td>1.3230 %</td>
<td>1.5320 %</td>
</tr>
</tbody>
</table>

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). The ROC charts are on the next page.
VeriEye 11.2 SDK matching engine with iris templates from IRE-2005 Exp1 (Right Iris) database (captured in near-infrared spectrum):
- Maximized matching speed scenario
- Maximized matching accuracy scenario

VeriEye 11.2 SDK matching engine with iris templates from ND-IIRS-0405 database, University of Notre Dame (captured in near-infrared spectrum):
- Maximized matching speed scenario
- Maximized matching accuracy scenario
VeriEye 1.2 SDK matching engine with iris templates from IRISDB1600 database, University of Bath (captured in near-infrared spectrum):
- Maximized matching speed scenario
- Maximized matching accuracy scenario

VeriEye 1.2 SDK matching engine with iris templates from MobileirisV1.0 DB, Warsaw University of Technology (captured in visible light spectrum):
- Maximized matching speed scenario
- Maximized matching accuracy scenario
VeriEye Demo, Trial SDK and Related Products

VeriEye algorithm demo application and VeriEye 30-day SDK Trial are available for downloading at www.neurotechnology.com/download.html.

These products are related to VeriEye SDK (see the corresponding product brochure for more information):

- **MegaMatcher SDK** – intended for development of AFIS or multi-biometric iris, fingerprint, face and voice identification products.
- **MegaMatcher On Card SDK** – a product for iris, fingerprint and face matching on smart cards.
- **NCheck Bio Attendance** – an end-user employee attendance management application designed as ready-to-use time and attendance system with biometric iris, face and fingerprint identification; VeriEye iris recognition algorithm is used in the application for checking person identity.
Licensing VeriEye SDK

Product Development

An integrator should obtain either a VeriEye 11.2 Standard SDK (EUR 339) or VeriEye 11.2 Extended SDK (EUR 859) to develop an end-user product based on VeriEye technology. The SDK needs to be purchased just once and may be used for all projects and by all the developers within the integrator’s company.

See the “Contents of VeriEye Standard SDK and Extended SDK” chapter (page 4) for the list of component licenses included with VeriEye 11.2 Standard and VeriEye 11.2 Extended 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 of components for every computer or device, where component will be installed together with integrator’s product. See Product Advisor to find out what specific components will be needed for the deployment of your system. Integrators can purchase additional VeriEye component licenses if required at anytime.

License activation options

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

- **Serial numbers** are used to activate licenses for particular VeriEye components on 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 ARM-Linux, except BeagleBone Black and Raspberry Pi 3 devices.
  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 particular VeriEye components 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 VeriEye components in the following ways:
  
  1. **Activating single computer licenses** – An installation license for a VeriEye 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 VeriEye components 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 a VeriEye component may be used as a dongle, allowing the VeriEye 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 ([https://www.neurotechnology.com/mm_110_sla.html](https://www.neurotechnology.com/mm_110_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 VeriEye-based development tool (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 VeriEye enterprise license allows an *unlimited use* of VeriEye components 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 for VeriEye Products

- The prices are **effective March 26, 2019**. The prices may change in the future, so please **download and review the latest version** of the brochure before making an order.
- 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.

<table>
<thead>
<tr>
<th>VeriEye SDK</th>
<th>VeriEye 11.2 Standard SDK</th>
<th>€ 339.00</th>
<th>VeriEye 11.2.1 Standard SDK</th>
<th>€ 859.00</th>
</tr>
</thead>
</table>

### Iris components for desktop (prices per single computer license)

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Iris Client (1)</th>
<th>Iris Extractor</th>
<th>Iris Matcher</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - 9</td>
<td>€ 38.00</td>
<td>€ 30.00</td>
<td>€ 38.00</td>
</tr>
<tr>
<td>10 - 19</td>
<td>€ 28.00</td>
<td>€ 22.00</td>
<td>€ 28.00</td>
</tr>
<tr>
<td>20 - 49</td>
<td>€ 25.00</td>
<td>€ 19.00</td>
<td>€ 25.00</td>
</tr>
<tr>
<td>50 - 99</td>
<td>€ 22.00</td>
<td>€ 17.00</td>
<td>€ 22.00</td>
</tr>
<tr>
<td>100 - 199</td>
<td>€ 19.00</td>
<td>€ 15.00</td>
<td>€ 19.00</td>
</tr>
<tr>
<td>200 - 499</td>
<td>€ 17.00</td>
<td>€ 13.00</td>
<td>€ 17.00</td>
</tr>
<tr>
<td>500 - 999</td>
<td>€ 15.00</td>
<td>€ 12.00</td>
<td>€ 15.00</td>
</tr>
<tr>
<td>1000 - 1999</td>
<td>€ 13.00</td>
<td>€ 11.00</td>
<td>€ 13.00</td>
</tr>
<tr>
<td>2000 - 3999</td>
<td>€ 12.00</td>
<td>€ 10.00</td>
<td>€ 12.00</td>
</tr>
<tr>
<td>4000 - 7999</td>
<td>€ 11.00</td>
<td>€ 9.00</td>
<td>€ 11.00</td>
</tr>
</tbody>
</table>

8000 and more Please contact us for more information

### Embedded iris components for Android, iOS and ARM Linux (prices per single computer license)

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Mobile Iris Client (1)</th>
<th>Mobile Iris Extractor</th>
<th>Mobile Iris Matcher</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - 9</td>
<td>€ 25.00</td>
<td>€ 20.00</td>
<td>€ 25.00</td>
</tr>
<tr>
<td>10 - 19</td>
<td>€ 18.00</td>
<td>€ 15.00</td>
<td>€ 18.00</td>
</tr>
<tr>
<td>20 - 49</td>
<td>€ 16.00</td>
<td>€ 13.00</td>
<td>€ 16.00</td>
</tr>
<tr>
<td>50 - 99</td>
<td>€ 14.00</td>
<td>€ 11.00</td>
<td>€ 14.00</td>
</tr>
<tr>
<td>100 - 199</td>
<td>€ 12.50</td>
<td>€ 10.00</td>
<td>€ 12.50</td>
</tr>
<tr>
<td>200 - 499</td>
<td>€ 11.00</td>
<td>€ 9.00</td>
<td>€ 11.00</td>
</tr>
<tr>
<td>500 - 999</td>
<td>€ 10.00</td>
<td>€ 8.00</td>
<td>€ 10.00</td>
</tr>
<tr>
<td>1000 - 1999</td>
<td>€ 9.00</td>
<td>€ 7.00</td>
<td>€ 9.00</td>
</tr>
<tr>
<td>2000 - 3999</td>
<td>€ 8.00</td>
<td>€ 6.40</td>
<td>€ 8.00</td>
</tr>
<tr>
<td>4000 - 7999</td>
<td>€ 7.00</td>
<td>€ 5.80</td>
<td>€ 7.00</td>
</tr>
</tbody>
</table>

8000 and more Please contact us for more information

### License management

| Volume license manager | € 16.00 |

(1) These components are not available for VeriEye Standard SDK customers.

VeriEye products can be ordered:

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