Allied Vision

RELEASE NOTES

eVision 26.10

2026-Oct-30



NOTICE

Starting with release 26.10, the **Open eVision** product has been renamed to **eVision** and will appear under the **Allied Vision** brand.



ONLINE DOCUMENTATION

The online eVision documentation is available at the Allied Vision documentation website: evision.docs.alliedvision.com

RELEASE BENEFITS

eVision has been rebranded

Being part of the Allied Vision ecosystem, the software now comes in an Allied Vision branded design. Experienced users will see a redesigned user interfaces. The technical IDs and installation folders will also be updated to align with the Allied Vision product portfolio. However, all features and tools will remain the same. And starting with release 26.10, **Open eVision** has been renamed to **eVision**.





Improvements of the code readers

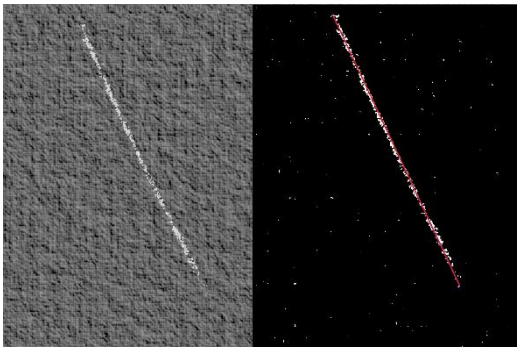
EasyBarCode now supports two additional symbologies: GS1 DataBar Stacked Omnidirectional and GS1 DataBar Expanded Stacked (see GS1 DataBar barcodes). An example demonstrating how to read these codes has been added to **eVision Studio**.

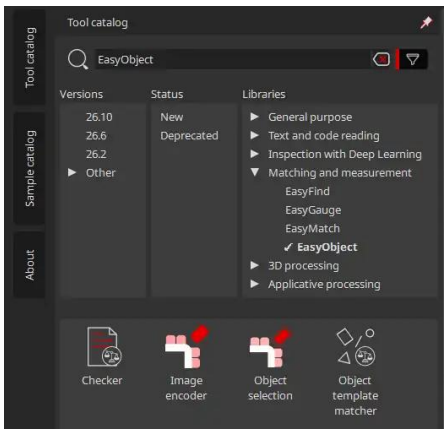
EasyQRCode reading performance has also been enhanced, with improved QR code grid positioning and more robust handling of uneven illumination and irregular grids.

Line detector in EasyImage

A new **EasyImage** line detector is available to detect straight line segments in BW8 binary images. It leverages a probabilistic Hough transform to identify candidate lines and then merges nearby collinear line segments. The ELineDetector class supports optional region and frame inputs and returns a list of ELine objects.

This feature is well suited for detecting scratches, cracks, and other straight marks on a wide range of materials. A corresponding tool and sample have been added to **eVision Studio**.





Usability improvements for eVision Studio

eVision Studio tool and sample catalogs now include a powerful filtering system. You can narrow the lists to items introduced in a specific **eVision** version, marked as New or Deprecated, or belonging to a particular library. Filters can be combined to further refine searches.

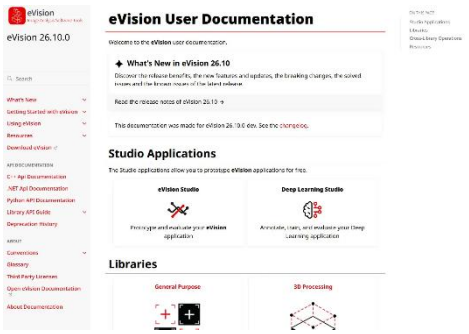
The **eVision** documentation is now available in a dockable window within the application.

New tools have also been added for **EasyOCR2**, Region to ROI, Frames from shapes, Custom convolution kernel, and acquiring images from a live video stream (RTSP).

eVision documentation overhaul

The **eVision** documentation portal has been fully redesigned.

With updated content, the new documentation brings together all the resources you need to build applications using **eVision** libraries.



RELEASE SPECIFICATIONS

OS and processor architectures

WINDOWS OS

- **eVision** is a 64-bit library that requires a processor compatible with the SSE4 instruction set.
- **eVision** runs on the following Windows operating systems, x86_64 architecture only:

OS version
Windows 11®
Windows 10® (version 1809 or above)
Windows 11® IoT Enterprise
Windows 10® IoT Enterprise

LINUX OS

- **eVision** is compatible with x86_64 and aarch64 (ARMv8-A) CPUs.
- **eVision** and the **Neo License Manager** are designed to be distribution-independent on x86_64 platforms.
- The minimum requirements are:
  - glibc version 2.31 and
  - libstdc++6 version 10.
- This release has been validated with the following distributions:
  - For deb based distributions:
    - Ubuntu 22.04, Ubuntu 24.04 and Ubuntu 26.04;
    - Debian 11, Debian 12 and Debian 13
  - For rpm based distributions:
    - Rocky Linux 10.
- This release has been validated on the following embedded systems:
  - Raspberry Pi 4, Raspberry Pi 5 and Raspberry Pi Zero 2 W.
  - NVidia Jetson Orin AGX with Jetpack 6 and Jetpack 7, and NVidia Jetson Orin NX with Jetpack 7 only.



PTHREAD LIBRARY DEPENDENCY ON LINUX

- For CMake samples: the client programs must be linked against the pthread library.
- For Qt samples: it is not necessary to make this dependency explicit because a program using Qt automatically depends on the pthread library.

NVIDIA GPU SUPPORT

- **eVision** uses **CUDA** v12.9 and supports only GPU with a compute capability of 7.5 or more: from architecture Turing (RTX2000 series) to Blackwell (RTX 5000 series). On aarch64, **eVision** only supports the Jetson Orin series.

REMOTE CONNECTIONS AND VIRTUAL MACHINES

- Remote connections:
  - You can install and use licenses on a remote connection using remote desktop, TeamViewer or any other similar software.
- Virtual machines:
  - Virtual machines are supported. Microsoft Hyper-V, Oracle VirtualBox and libvirt hypervisors have been successfully tested.



WARNING

OPENGL DRIVER LIMITATIONS ON VIRTUAL MACHINES

Depending on the configuration and the OpenGL driver available on your virtual machine, the Easy3D::E3DViewer class, the 3D tools in eVision Studio, or some of their features might be inoperative.

- Regarding license management, only the Neo Licensing System is compatible with virtualization.

Supported programming languages and IDEs

The **eVision** libraries and tools support C++, Python and the programming languages compatible with .NET (C#, VB.NET).

C++ REQUIREMENTS

- A compiler compatible with the C++ 11 standard is required to use **eVision**.
- Some **eVision** samples use the Qt framework to create a user interface and display images with a graphical overlay. The recommended Qt versions are 5.12 to 5.15.

.NET REQUIREMENTS

- The .NET framework 4.8 (or later) or the .NET platform 6.0 (or later) is required to use the C# version of **eVision**.
- The .NET version of **eVision** is exclusively available on Windows.

PYTHON REQUIREMENTS

- Python 3.11 or later is required to use the Python bindings for **eVision**.
- No other third party dependencies are necessary to use the Python bindings.

COMPATIBILITY WITH IDES

- Any Windows and Linux IDE supporting a compatible compiler/environment. The recommended ones are:
  - C++: MSVC 2022 or higher, g++ 11 or higher
  - .NET: .NET 6.0 or higher
  - Python: Python 3.12 or higher

CHANGELOG FOR 26.10

Specification Changes

- Adds Debian 13 and Ubuntu 26.04 to the list of tested Linux distributions. Also replaces RHEL 10 by Rocky Linux 10.
- Adds Jetpack 7 to the list of tested aarch64 platforms and cites exact Jetson machines that were tested.
- Now clearly specifies aarch64 platforms where EasyDeepLearning is supported. (Jetson Orin series only.)
- Now specifies `libstdc++6` exact version dependency instead of `gcc` version for Linux system dependencies.
- Upgrades minimal system requirements for Linux to `glibc` version 2.31.
- Now clearly specifies recommended compiler/environment versions. Those are distinct from minimal requirements from compilers. Those recommended versions are: MSVC 2022, g++ 11, .NET 6.0 and Python 3.12.
- Now clearly specifies that the .NET version of **eVision** is Windows-only. (It was always the case as we never provided Linux .NET bindings but was not indicated in the specifications.)

eVision Studio

- New features
  - At the top of the Tool Catalog, a new keywords menu now appears below the search field in order to propose keywords to help filling it.
  - At the top of the Sample Catalog, a new keywords menu now appears below the search field in order to propose keywords to help filling it.
  - The eVision documentation is now accessible in a dockable window.
  - New samples:
    - Reading of Gs1 Databar Stacked barcodes
    - EasyOCR2 reading with a deep learning classifier.
    - EasyOCR2 reading using a character database.
    - Reading digits across a point gauge
  - Region Morphology tool: added a new `EFrame` input
  - Region Transform tool: added a new `EFrame` input
  - EasyLocate tool: added a new `EFrame` input and a new `ERegion` input (mask)
    - Output can be Array of `ELocatorPredictedObject` as well
  - EasySupervisedSegment tool: added a new `EFrame` input and a new `ERegion` input (mask)

        - Output can now be Array of `ESupervisedSegmenterBlob` or `EObjectSelection` as well
    - Convolution tool: can now take an `EKernel` as input to perform an `EasyImage::ConvolKernel`.
    - ToRegion tool:
        - can now take `ESupervisedSegmenterBlob` and `ELocatorPredictedObject` as input
        - can now take `EOCR2Text` as input
        - can now take `ECircle`, `EPolygon`, `ERectangle` and `EWedge` as input
    - ToFrame tool:
        - can now take `ELocatorPredictedObject` as input
        - can now take `EOCR2Text` as input
    - New tool added:
        - EasyOCR2 reader.
        - Region To ROI tool that that takes an `ERegion` and an optional padding to construct a ROI from the bounding box of the input `ERegion`.
        - Create frames on shape.
        - Kernel convolution tool to define a custom convolution kernel.
        - Stream source tool that connects to an RTSP stream (via a locally installed GStreamer) and exposes the latest decoded frame as an image output.
    - Checker tool: now has a Draw Blobs checkbox to choose whether to draw found blobs
    - Region Combiner tool: can now take an Array of ERegion as input and combines all its regions together
    - ToArray and FromArray tools: now support `ELine`, ECircle, ERectangle, EPolygon, EWedge and EGrid items and arrays, and draw the selected item on their displays.
- Changes
    - eVision Studio is now Allied Vision-branded software. Its style was updated to use Allied Vision colors and logo.
    - The project file extension has been changed from `.oes` to `.evs`. Existing `.oes` projects can still be loaded.
    - The C++ generated code was updated to only include the necessary headers.
    - Handling of image and zoom was optimized to make the application smoother.

- The slider that occasionally appears at the right of a SpinBox (number field) is now located below it.
- Draw 2D tool: It is now possible to set the alpha value for the pen color.
- Region Editor tool: region primitives are now created on the current view rather than in the middle of the image.
- Region editor, Region to image, Frame, Point and Shape (Circle, Line, Polygon, Rectangle, Wedge) tools: the surface/image size setting now enforces a common guideline (minimum 1x1, maximum 64000x64000, default 1920x1080 pixels).
- Bearing Inspection sample: the sample now inspects the 16 rolling elements using the new CreateFramesOnShape tool.
- The ColorPickerButton now have a checkerboard background. This makes the alpha value of the color visible.
- The Draw2D tool now disables the Brush when setting it has no influence on the drawing. It was also removed from generated code in these cases.
- Draw 2D tool: It is no longer possible to change the style and the width of the pen when the input is an `ERegion`. This previously did nothing since the `ERegion::Draw` and `ERegion::DrawContour` method only take a color as argument.
- `EPoint::Draw` is now used instead of `EPoint::DrawHandle` in the Draw2D, Point and ObjectTemplateMatcher tools.

- Solved issues
  - eVision Studio would crash when a Draw2D tool with the “Features” checkbox checked was connected to the output of a Matcher tool. This has been corrected.
  - A crash could occur on Windows when loading a project with 50+ links in the graph. This has been fixed.
  - The generated code for the Threshold tool when using a relative threshold and a `EImageBW16` was wrong. This has been fixed.
  - The Wedge Tool generated a wrong code for the breadth value in the constructor, this has been fixed.
  - eVision Studio would crash when the `ERegion` output of an Image To Region tool was connected to another tool. This has been fixed.
  - Frame, Point and Shape (Circle, Line, Polygon, Rectangle, Wedge) tools: pressing Reset did not restore the surface size to its default value. This has been fixed.
  - The Line tool could not be duplicated properly. This has been corrected.
  - The selected code/index parameter was not properly copied when the BarCodeReader, MailBarCodeReader, PatternFinder, Matcher, QRCodeReader and MatrixCodeReader tools were duplicated. This has been corrected.
  - The learned region parameter was not properly copied when the PatternFinder and Matcher tools were duplicated. This has been corrected.

- In the Object Template Matcher and Transform tool, a display widget could take all the space and compress the other. This has been fixed.
- In the Object Template Matcher tool, modifying the pan and zoom in the view widget updated the state of the setup widget. This has been fixed.
- Updating the parameters of the 3D Laser Line Object Calibration tool would not always update the 3D calibration model. This has been fixed.
- eVision Studio would crash when not enough memory was available to allocate the new image in the resize tool. This has been fixed.
- The python generated code for the QRCode Reader tool was missing a line break, this has been fixed.
- The python generated code for the EChecker tool was incorrect, this has been fixed.
- The python generated code for the Region To Image tool was incorrect, this has been fixed.
- The C# generated code for the BarCode Reader tool was incorrect, this has been fixed.
- The C# generated code for the White Balance tool was incorrect, this has been fixed.
- The generated code for the EPointGauge tool was incorrect, this has been fixed.

Deep Learning Studio

- New features
  - Chinese translation of Deep Learning Studio.
- Changes
  - Deep Learning Studio is now Allied Vision-branded software. Its style was updated to use Allied Vision colors and logo.
    - The default path for the projects is now `[Documents]/eVision/Deep Learning Studio Projects`. Previously created projects will remain in their old folder and will still be accessible in this new version.
    - The configuration file of Deep Learning Studio is now saved in an `Allied Vision` folder. The old configuration file will be automatically imported the first time you open Deep Learning Studio. If you use older versions of Deep Learning Studio, you will have two separate copies of the Deep Learning Studio configuration file.
  - It is now possible to do a benchmark with any EasyDeepOCR tool.
  - The main menu of Deep Learning Studio has been reworked and now leave more place to the tabs.
  - Pressing the Control key while annotating an EasyLocate project would switch to Add mode while pressed, and back to Edit mode when lifted. This could cause issues when zooming, as it required pressing the same button. The button to switch annotations was changed to the Shift key.
  - When creating a benchmark, we no longer ask for the number of images, but rather the number of inferences.
- Solved issues

- A freeze could happen when trying to optimize the batch size of a benchmark, while something was running in the background. This has been fixed.
- Loading a 16-bit image could fail with the message “An unknown error occurred while importing images to the dataset”. This has been fixed.
- EasySegment Supervised and Unsupervised benchmark could in some instance be wrong. This has been fixed
- Removing labels could make the application crash. This has been fixed.
- Importing an “EasyDeepLearning dataset” in an EasySegment Supervised or EasyLocate project that already contains some labels could lead to the imported annotation to be assigned the wrong label. This has been fixed.

Across All Libraries

- Changes in all versions (C++, DotNet, Python)
  - The unused enum values named `reserved` + a number present in many enumerations have been removed. ⚠ **breaking change** ⚠
- Changes in C++ Version
  - All the types of the eVision library can now be included through smaller header files using `#include <eVision/<sub_library>/<type_name>.h`. The previous recapitulative headers (like `Easy.h` or the global header `eVision.h`) remain usable, but it is now recommended to use the small headers to optimize compilation speed in the C++ version. The necessary include lines for each type are cited in the new Doxygen API documentation published along the eVision documentation.
  - The type `OEV_INT64` has been removed. All instances of this type can be replaced by `int64_t`. ⚠ **breaking change** ⚠
  - The type `OEV_UINT64` has been removed. All instances of this type can be replaced by `uint64_t`. ⚠ **breaking change** ⚠
  - The type `OEV_INT32` has been removed. All instances of this type can be replaced by `int32_t`. ⚠ **breaking change** ⚠
  - The type `OEV_UINT32` has been removed. All instances of this type can be replaced by `uint32_t`. ⚠ **breaking change** ⚠
  - The type `OEV_INT16` has been removed. All instances of this type can be replaced by `int16_t`. ⚠ **breaking change** ⚠
  - The type `OEV_UINT16` has been removed. All instances of this type can be replaced by `uint16_t`. ⚠ **breaking change** ⚠
  - The type `OEV_INT8` has been removed. All instances of this type can be replaced by `int8_t`. ⚠ **breaking change** ⚠

- The type `OEV_UINT8` has been removed. All instances of this type can be replaced by `uint8_t`. ⚠ breaking change ⚠
- The type `euresys_smart_ptr` has been removed. All instances of this type can be replaced by `std::unique_ptr`. ⚠ breaking change ⚠
- The type `BOOL` and the values `TRUE` and `FALSE` have been removed from the eVision headers. When using the eVision API, you should adapt these to `bool`, `true` and `false`. You can also rely on automatic conversion rules and use `int`, `1`, and `0`. Note that this type and these values are also used and defined by the Windows API. In those cases, you should keep the usage of `BOOL`, `TRUE` and `FALSE`. ⚠ breaking change ⚠
- The unused enum value `<name_of_enum>_reserved` always present in all enums have been removed. ⚠ breaking change ⚠
- Removes legacy tolerance for non-UTF8 encoded `std::string` in the methods taking strings as input that was applied only on Windows platform. Now the Windows version will behave exactly like the Linux version and throw a `EException` with error code `EError_InvalidUTF8Encoding` if any `std::string` containing non-valid UTF-8 is passed to a method. ⚠ breaking change ⚠
- Removes auto-include mecanisms based on `#include` ordering to include `eVision::EGrabberBridge` and `eVision::VimbaXBridge` namespaces. Now, in order to use the classes in these namespaces, the `#include <eVision/EGrabberBridge.h>` and `#include <eVision/VimbaXBridge.h>` include lines must be used respectively. ⚠ breaking change ⚠

Easy

- New features
  - Adds `Easy::EBaseROI::IsAttached` method as a replacement for some use cases of the now deprecated `GetParent` method.
  - Adds `Easy::EBaseROI::GetBaseImage` method as well as the `GetImage` method as drop-in replacements for the now deprecated `GetBaseTopParent` and `GetTopParent` methods.
  - `EIntegerRange` now has an equality and an inequality operator.
  - `EFloatRange` now has an inequality operator.
  - Adds `ERegion::FillHoles()` to return a new `ERegion` whose holes have been filled.
  - Adds `EFoundPattern::GetHasFeaturePoints()` to query if the `EFoundPattern` contains feature points.
  - Adds `CreateFramesOnShape` to `ECircle`, `EWedge`, `ERectangle`, `EPolygon` and `ELine` to create a vector of frames on the perimeter of the shapes.

- Adds `EGrid::CreateFramesOnGrid` to create a vector of frames in the center of each cell of the grid.
- Adds an overload to `EBaseROI::SetPlacement` and `EBaseROI::Attach` that takes an `ERegion` and an optional padding to construct a ROI from the bounding box of the input `ERegion`.
- Adds 4 new constructors to `ERegion` to construct it from an `ERectangle`, `EPolygon`, `ECircle` or `EWedge`.
- Add a new constructor to `EPolygonRegion` to construct it from an `EPolygon`.
- `Easy::DegreesToRadians` and `Easy::RadiansToDegrees` allow conversion of angles between degrees and radians units.
- In `EWindowsDrawAdapter` and `ESystemDrawingDrawAdapter`, added the facultative parameter `enhancedDisplay` in all constructors.

- Changes
  - The structs `EC15`, `EC16`, `EBW1` have been removed. **⚠ breaking change ⚠**
  - The "Euresys simplified format" was renamed to "eVision simplified format".
- Deprecations
  - The following methods in `EBaseROI` and all its subclasses are now deprecated: `GetParent`, `HasSubROI`, `GetFirstSubROI`, `GetNextSiblingSubROI`, `GetNextROI`, `GetNextSiblingROI`, `GetSubBaseROIs` and `GetSubROIs`. The motivation is the high runtime cost on all operations associated with the maintenance of a hierarchy of ROIs within the class as well as the low usage of that feature. For users using that feature it is recommended to re-implement it on top of those classes.
  - The `GetBaseTopParent` and `GetTopParent` methods in `EBaseROI` and all its subclasses are also deprecated but have drop-in replacements: `GetBaseImage` and `GetImage`
  - `Easy::SetAngleUnit`, `Easy::GetAngleUnit`, `Easy::FromRadians`, `Easy::ToRadians`, `Easy::FromDegrees` and `Easy::ToDegrees` are deprecated and will be removed in 27.10. After the end of the deprecation period, changing the angle unit will no longer be possible: all angles supplied to and returned by **eVision** will always be in degrees. Code generated by eVision Studio no longer calls `Easy::FromDegrees`.
  - The `Easy::GetEnableEnhancedImageDisplay()` and `Easy::SetEnableEnhancedImageDisplay()` methods are deprecated in favor of passing these parameters in the constructor or `EWindowsDrawAdapter` instead.
- Solved issues
  - Copied regions were not thread safe and a heavy usage of a region and a copy in two different threads could lead to crashes. This has been fixed.

- Errors during calibration (automatic or explicit) could make eVision crash or would be ignored. Now, `ECameraCalibration::Calibrate` will always throw when an error occurred during the calibration (for example less than 8 landmarks for the advanced mode or landmarks that give a singular matrix) and `ECameraCalibration::AutoCalibrate` will throw only when all the tested mode yielded an error.

EasyImage

- New features
  - Adds `EasyImage::FillHoles()` that takes a binary `EImageBW8` or a `ERegion` and returns a binary `EImageBW8` whose holes have been filled.
  - Adds `EasyImage::RemoveOutliers()` which replaces a pixel by the median of the pixels in the surrounding if it deviates from the median by more than a certain value (the threshold). Useful to remove noise and fill small holes.
  - Adds `EasyImage::Thinning()` that takes a binary `EImageBW8` or a binary `EImageBW16` and returns a binary image of the same type containing a 1 pixel-wide skeleton representing the overall shape of the input image.
  - Adds `ELineDetector` that takes a binary `EImageBW8` and returns a list of straight lines found in the input image.
  - Adds Copy constructor, assignment operator and serialization to `EKernel`.
- Solved issues
  - ConvolGaussian now works properly when halfKernelWidth is >= 64.

EasyBarCode

- New features
  - Added support for 2 new symbologies EBarCodeSymbologies_Gs1DataBarStackedOmnidirectional and EBarCodeSymbologies_Gs1DataBarExpandedStacked.
- Changes
  - Reading capabilities were slightly improved, especially for Gs1 Databar symbologies.
  - `EBarCodeSymbologies_Gs1DataBar` and `EBarCodeSymbologies_Rss14` Codes whose payload is within [20000000000000, 20585067703929 [were previously incorrectly decoded and are now ignored.
- Solved issues
  - `EBarCodeSymbologies_Gs1DataBarExpanded` and `EBarCodeSymbologies_Rss14 Expanded` Codes containing 11 finder patterns (the maximum allowed by the symbology) were not decoded correctly. This has been corrected.
  - In rare cases, the corners of `EBarCode::GetPosition` were not correct.

EasyQRCode

- Changes
  - Overall improvement of QRCode reading capabilities (better positionning of QRCode grid, better handling of irregular illumination, and better handling of irregular grids).
  - Users can now set the `AdaptiveThresholdHalfKernelSize` of their `EQRCodeReader`. This setting was previously computed from the input image size, resulting in instabilities in complicated cases when using ROIs. The `AdaptiveThresholdHalfKernelSize` is now `50` by default, this works better than the previous setting overall but regressions are possible in some cases. If this concerns you, you may restore the previous behaviour by setting `ComputeAdaptiveThresholdHalfKernelSizeFromImage` to `true` or tune the parameter manually. **⚠ breaking change ⚠**
- Solved issues
  - In some rare cases, reading could throw an `EException` of type `EError_PatternTooSmall`, this has been fixed.

EasyOCR2

- New features
  - A `ToRegion` and a `ToFrame` method has been added to `EOCR2Text`.
  - New method `EOCR2::Read2` that returns a `EOCR2Text`.
- Changes
  - `DrawSegmentation` was unable to correctly draw the segmentation when the zoom factor was different from 1.0 or when a frame was used when reading and drawing.
  - Changing the angle unit between after calling `EOCR2::SetTextAngleRange` would render the internal angle range of `EOCR2` invalid, messing the subsequent reads and call to `EOCR2::GetTextAngleRange`. This fix may implies setting the angle range of serialiazed EOCR2 object saved before this release.
  - In some specific cases, when a `EOCR2::Detect` failed to detect the actual texts, the subsequent called to `EOCR2::Recognize` could make eVision crash (no validate candidate for a falsely detected character could be found). Instead, a proper exception explaining the problem is now thrown.
  - When using `EOCR2::Detect` with a ROI and a `ERegion` and when the `ERegion` had to be cropped to fit inside the given ROI, the segmentation was either wrong and/or eVision could crash.
  - The `GetBoundingBox` method of `EOCR2Char`, `EOCR2Word`, `EOCR2Line` and `EOCR2Text` now returns a bounding box with the correct angle suitable to build a `EFrame`. More precisely, a text that is not oriented and read from left to right returns a bounding box with an angle of 0°.

- `EOCR2::GetCharsWidthRange`, `EOCR2::GetRelativeSpacesWidthRange` and `EOCR2::GetTextAngleRange` previously returned a reference (to be used as a setter). It is no longer the case.
  Use `EOCR2::SetCharsWidthRange`, `EOCR2::SetRelativeSpacesWidthRange` and `EOCR2::SetTextAngleRange` instead. ⚠ **breaking change** ⚠

- Deprecations
  - `EOCR2::ClearResult`, `EOCR2::GetReadText` have been deprecated. In the future, the class EOCR2 will no longer hold the last read result.
  - `HitTest` and `Draw` methods previously attached to the class `EOCR2` have been deprecated. Use the `HitText` and `Draw` methods of the returned `EOCR2Text`.
  - `EOCR2::Read` method has been deprecated. Use the new `EOCR2::Read2` that returns a `EOCR2Text`.
  - `EOCR2::Recognize` methods which takes a `const EOCR2Text&` as input and returns a `std::string` have been deprecated. Use the new `RecognizeText` methods that modifies the `EOCR2Text` instead.
  - Previously, `EOCR2::SetGlobalSegmentationThresholdMode` could accept a positive integer instead of a proper `EThresholdMode` (done through a wrong cast and from the fact that possible enum values for `EThresholdMode` were all negative) and used this integer to set the `GlobalSegmentationAbsoluteThreshold`. This still works for legacy code, but the new `SetGlobalSegmentationAbsoluteThreshold` should be used instead. This behavior is therefore considered deprecated.

EasyDeepOCR

- Changes
  - User retraining of the recognition network was improved: the model is now able to learn more complex use cases but the training speed will be slower, especially for very small datasets.

- Deprecations
  - `ETextReader::HasLocationCustomModel`, `ETextReader::HasRecognitionCustomModel` and `ETextReader::HasOrientationCustomModel` have been deprecated.
    Use `GetHasLocationCustomModel`, `GetHasRecognitionCustomModel` and `GetHasOrientationCustomModel` instead.

- Solved issues
  - When training an EasyDeepOCR recognizer, in some rare instances with very narrow font or with bad annotations, a crash could occur. This has been fixed.

EasyObject

- New features

- `ECodedElement` and `ECodedImage2` now have a `SetFrame` method. `ECodedImage2` also has a `GetFrame` method. It is useful to retrieve the frame after an encoding made with a frame or set the frame in after the encoding in case a frame was not given.

- Solved issues
  - Encoding a `EImageBW16` with an `EImageEncoder` with another encoder than `ESegmentationMethod.LabeledImage` made eVision crash. This has been fixed.

Deep Learning Inspection Libraries

- New features
  - A new class, `EDataset`, was introduced to replace `EClassificationDataset`. Its API is largely compatible with `EClassificationDataset`, except for the data augmentation settings that were removed. The data augmentation settings should be handled through `EDataAugmentation`.
  - Existing Deep Learning snippets have been moved from the `Deep Learning Tools` page to their corresponding `EasyClassify` and `EasySegment` pages.
  - TensorRT Engine
    - DLA support has been added to the TensorRT engine. This feature currently supports EasyClassify and EasySegment networks only. DLA devices can now be used for inference on NVIDIA Jetson Orin hardware. 👁 **Preview** 👁
  - EasyLocate
    - New snippets available: Creating a Dataset and Training a Locator and Loading a Locator and Locating Objects on a New Image.
    - Added a ToRegion() method to `ELocatorPredictedObject` and `ELocatorResult`.
    - `ELocator` now supports frame when computing results.
    - `ELocatorPredictedObject` now has a non-null angle when the result has been computed with an `EFrame`.
  - EasySegment
    - Added a ToRegion() method to `ESupervisedSegmenterResult` and `ESupervisedSegmenterBlob`.
    - Added a GetCodedImage() method to `ESupervisedSegmenterResult`.
    - `ESegmentSupervised` now supports frame when computing results.

- Changes

- `EDeepLearningProject::GetDataset()` now returns an `EDataset` object instead of an `EClassificationDataset` object. ⚠ **breaking change** ⚠
- The `EDeepLearningBenchmarkSettings::Set/GetNumImages` methods were removed. Use `EDeepLearningBenchmarkSettings::Set/GetNumInferences` for similar behavior, by interacting with the number of iterations of the benchmark instead. (see Deep Learning Studio change for more explanations.) ⚠ **breaking change** ⚠
- When choosing to optimize the batch size of a benchmark, it is now done during the benchmark and thus can change between two runs.
- Deep Learning Additional Resources
  - Added `.edldataset` files to all existing dataset in the Deep Learning Additional Resources. All datasets can now be loaded directly with their annotations in a custom Deep Learning project.
  - The `EasyLocate` directory was split into `EasyLocate - Axis Aligned Bounding Box` and `EasyLocate - Interest Point`.

- Deprecations
  - `EClassificationDataset` has been deprecated. Use `EDataset` instead.
  - The enumeration `EClassifierCapacity` has been deprecated. All methods using this enum were already deprecated.
    Use `EClassifier::GetModelType` and `EClassifier::SetModelType` instead.

Samples

- Changes
  - For all MSVC samples, opening files now places you by default in the most appropriate folder of the Sample Images or in the last folder you opened.
  - The CMake, MsVc and Qt samples as well as the C++ demo applications now only include the necessary headers. This results in ~20% faster build times.
  - The CMake samples now handle exceptions cleanly, such as missing licenses.
  - The Qt samples now build using Qt6 instead of Qt5.
  - The embedded samples using libcamera are now compatible with the latest release of libcamera.

Documentation

- New documentation portal
  - evision.docs.alliedvision.com is the new portal for the **eVision** documentation
- Code snippets
  - The C++ code snippets were updated to only include the necessary headers.

Licensing

- Solved issues
    - The **Neo License Manager** installed with the Linux RPM packages did not work properly. This has been fixed.

KNOWN ISSUES

System and Installation

MEMORY LEAKS

- If you use the CRT library to detect memory leaks in your program, it can falsely detect some memory leaks when you use the library.
    - This is a known limitation of the CRT library memory leak detection scheme. See Microsoft documentation.
    - It happens when the memory leak detection scheme is ended before the DLL is unloaded or the code in the headers is uninitialized.

EVISION INSTALLER

- There is a conflict between the eVision installer and any program using the UDP:6001 port. When a software is already using this port, the installation fails and rolls back.
    - **Workaround**: Install eVision first, and then the other software.



HINT

- This port is typically used by National Instrument software such as LabView.

Libraries

ALL

- Since version 24.02, cannot read back the serialization of some objects made with previous versions.
    - `ERectangleRegion` or E3DObject are known to be affected by this issue.

IMAGE FILE FORMATS

- TIFF files containing RGB values + alpha values are not supported.

EASYIMAGE

- The constructor `EKernel` using an `EKernelType` as parameter does not use the same rectifier as the associated method Convol.
    - For instance, `EKernel(EKernelType_GradientX)` returns a kernel with a `EKernelRectifier_KeepPositive` rectifier while the method `ConvolGradientX` uses a `EKernelRectifier_Absolute` rectifier.
- The method `ConvolKernel` does not have the same border value behavior (constant) as the pre-defined method like `ConvolGradientX` which is mirror.
- `EHarrisInterestPoints` can be slow to draw when there are many results.

EASYOBJECT

- `ECodedImage2` and `EObjectSelection` can be slow to draw when there are many results.

EASYMATCH

- Matching a vertically symmetric pattern with an angle tolerance around 180° and in the original image can lead to an error of 1 pixel on the detected position.
- By default, EasyMatch interpolation does not work on 15 x 15 and smaller patterns.
    - **Workaround**: For pattern sizes smaller than 16 x 16, adjust the `MinReduced` area to fit the `MinReducedArea` < W*H/4 (if interpolation is needed).

EASYGAUGE (LEGACY API)

- In .NET, the `EPointGauge.GetMeasuredPoint()` overload with no argument is not available. To get the default measured point, use -1 as index.
- By design, an `ELineGauge`, `ERectangleGauge`, `ECircleGauge` or `EWedgeGauge` is reported as invalid if at least one of its sample points is invalid. In addition, these invalid sample points cannot be drawn as they have not been measured successfully.
- The `EWedgeGauge::SetActiveEdges()` method incorrectly gets the `EDragHandle_Edge_r` and `EDragHandle_Edge_RR` bits mixed up when processing its argument.
    - **Workaround**: In order to activate the inner circle, set the `EDragHandle_Edge_RR` flag and use the `EDragHandle_Edge_r` flag to activate the outer circle.
- Using a gauge on an ROI leads to drawing problems.
    - **Workaround**: Use the gauge on the parent image.
- In the custom `EDraggingMode_ToEdges` dragging mode, you cannot resize the nominal wedge gauge position using the on-screen handles, neither in a custom application nor in or in Eval.
    - **Workaround**: Enter numerical values for the wedge gauge position.
- When attaching a gauge to a worldshape or another gauge, you must keep a reference of this gauge alive (that is keep the gauge in a variable or an array in your program) otherwise it “disappears” from the hierarchy (immediately in C++ , depending on the garbage collector in Python and C#).

EASYMATRIXCODE

- EasyMatricCode1
    - When grading is enabled, the optimizations are made in order to get accurate grading rather than have the best possible reading. As a result, the number of decoding errors reported with grading can be higher than without grading.
    - Inspecting images with a lot of details, even if they are low contrast, can require much more time spent in EasyMatrixCode than the TimeOut set previously.

DEEP LEARNING TOOLS

- Deep Learning tool objects (`EClassifier`, `EUnsupervisedSegmenter`, `ESupervisedSegmenter`, `ELocator`, `EDeepOCRTool`) will not free all the CPU and GPU memory that they allocated at destruction. This is due to third party libraries used for Deep Learning that maintain global caches or buffers.
    - So, it is not recommended to create and delete a lot of Deep Learning tools in the same program.

Licensing

LEGACY LICENSE MANAGER

- The **Legacy License Manager** might not start if the **.NET Framework 4.8** is not installed.
- Using in English language mode on a Chinese or Japanese Windows version can lead to truncated text being displayed. This is an issue linked to the automatic font selection and there is currently no workaround. Please note however that, by default, the runs in the OS language, including Chinese and Japanese.

NEO LICENSE MANAGER

- In some cases, offline to offline reactivation may not work.

LONG STARTING TIME

On some installations, the licensing system can take a long time to start (from 10 seconds up to a few minutes). If you have this issue, you can try the following procedures:

- Clean your software license cache.
    - The software license cache can become bloated by usage.
    - It can also happen if you use only dongles, as the system checks the presence of software licenses in all cases.
    - To clean the cache, use the LicenseManager.exe /DeleteLicenseFiles command.



WARNING

This command deletes all the licenses that are not managed by the on the system. Reactivate these licenses after the cleaning.

- Update your system root certificates.
    - If your root certificates are expired, the validation of the licensing system signatures might fail and timeout.
    - This only happens if the computer is on a network, even if the network is not connected to the Internet.
- Enable only the licensing system(s) you use.
    - By default, all the supported licensing systems are enabled.
    - Use the new (available from 2.13) Preconfiguration::SelectLicensingModels method to select exactly the licenses you want to enable and avoid issues arising from the usage of the other ones.

Samples

- In the provided Python samples using both and Qt, a rare bug was reported causing failure at loading shared library due to negative interaction with the Qt library on some computers only.
    - The conditions necessary to reproduce this bug are currently unknown.
    - A workaround and an explanation of this bug is now provided in those Python / Qt samples.

END OF LIFE AND DEPRECATIONS

Support of older environments

OS AND PROCESSOR ARCHITECTURES

OS	Architecture	Last version with support
Windows 8®	64-bit	Open eVision 25.06
Windows 7® (*)	64-bit	
Windows 11®	32-bit	Open eVision 23.04
Windows 10®	32-bit	
Windows 8®	32-bit	
Windows 7®	32-bit	
Windows Vista	—	Open eVision 2.3.3.10777
Windows XP	—	Open eVision 2.2.2.10255
Windows 2000	—	Open eVision 1.0.1.5222

(*) The recommended version is 6.1.7601 (Windows 7 Service Pack 1)

.NET FRAMEWORK

- Starting with 25.02 (February 2025 release) the minimum required version for the .NET Framework will be 4.8.

SUPPORTED IDES

IDE	Last version with support
Microsoft Visual Studio 2008	Open eVision 22.12
Microsoft Visual Studio 2005	Open eVision 2.7
Microsoft Visual Studio 2003	
Microsoft Visual Studio 6.0	Open eVision 2.5.1.1107

PROGRAMMING LANGUAGES

Programming language	Last version with support
Borland C++ Builder 6.0	Open eVision 2.5.1.1107
CodeGear Delphi 2009	

CodeGear C++ Builder 2009	
ActiveX API	
Embarcadero RAD Studio XE4 and XE5	Open eVision 2.4.1.11114
Borland Delphi 6.0 and 2006	Open eVision 1.0.1.5222
Borland C++ Builder 2006	

SOFT-BASED / HOST LICENSES

- The **eVision** soft-based / host licensing system will be progressively phased out.
    - It was introduced in 2007 with the first version of and it is based on an old, and now obsolete technology.
    - It is superseded by the **Neo Licensing System**.
    - The last version to support soft-based / host licenses is Open eVision 23.12.
- Milestones of the phase out period:
    - Since 1 January 2023:
        - There are no more sales of soft-based / host licenses.
        - The related product codes are 4250 to 4289.
        - These products are removed from the price lists.
    - Since 1 January 2024:
        - The support for the soft-based / host licenses is removed from the libraries.
        - The releases 24.02 and later are not be able to detect and use any soft-based / host licenses activated on the platform.
    - Starting 1 January 2029:
        - The soft-based license operation server will be shut down.
        - Activating or recovering a soft-based / host license will not be possible after 2028.



NOTE

- All applications using a soft-based / host license will continue to work “forever”, as long as the licenses have been activated and don’t require recovery.
- The Neo Licensing System replaces the soft-based / host licensing system.
- The Neo Licensing System supports dongles and for licenses.
- The usage and features of the Neo Software Containers are the same as the old soft-based / host licenses.
- Legacy eVision licenses are not affected by these changes.

LEGACY HEADERS

- The **Legacy Headers** are removed from **eVision** distributions since the end of 2022.
    - The **Legacy Headers** help the customers to migrate an existing application using **Legacy eVision** (last major release in 2006) to **eVision**.
    - The last version with support is **Open eVision 22.12**.

3D STUDIO

- **3D Studio** is removed from the **eVision** package since the end of 2025.
    - We recommend prototyping your 3D pipeline in **eVision Studio**.
    - The last version that includes the **3D Studio** application is **Open eVision 25.10.1**.

LEGACY OPEN EVISION STUDIO

- **Legacy Open eVision Studio** is removed from the **eVision** package since **Open eVision 26.6**.
    - We recommend prototyping with the current version of **eVision Studio**.
    - The last version that includes the **Legacy Open eVision Studio** application is **26.02.3**.

Deprecation History

DEPRECATION PROCESS

The deprecation of a feature is always announced in the release notes. The deprecated feature stays in the distribution (API, sample, eVision Studio…) for one year. After a year, it is completely removed from the distribution.

DEPRECATION LIST

*Deprecated in **26.10***

- EasyDeepLearning::EClassifierCapacity
    - Replaced by EClassifier::SetModelType and EClassifier::GetModelType
    - Will be removed in **27.6**
- EOCR2::ClearResult, EOCR2::GetReadText
    - Replaced by the returned EOCR2Text
    - Will be removed in **27.10**
- EOCR2::HitTest, EOCR2::Draw*
    - Replaced by the method attached to the returned EOCR2Text
    - Will be removed in **27.10**
- std::string EOCR2::Recognize(const EOCR2Text&)
    - Replaced by EOCR2::RecognizeText

  - Will be removed in **27.10**

- std::string EOCR2::Read()
  - Replaced by EOCR2::Read2
  - Will be removed in **27.10**
- EasyDeepLearning::EClassificationDataset
  - Replaced byEasyDeepLearning::EDataset
  - Will be removed in **27.10**
- Easy::FromRadians, Easy::FromDegrees, Easy::ToRadians, Easy::ToDegrees, Easy::SetAngleUnit and Easy::GetAngleUnit
  - After the end of the deprecation period, all angles given as parameters or arguments to eVision methods and classes will be expected in degrees instead of the unit specified by Easy::SetAngleUnit.
  - Will be removed in **27.10**
- EImageFileType_Euresys
  - Replaced by EImageFileType_eVision
  - Will be removed in **27.10**

*Deprecated in **26.6***

- EasySpotDetector library (ESpot and ESpotDetector classes)
  - Will be removed in **27.6**
- EObjectTemplateMatcher::EnableAlignment
  - Replaced by EObjectTemplateMatcher::Alignment
  - Will be removed in **27.6**
- EObjectTemplateMatcher::SortPositions
  - Replaced by EObjectTemplateMatcher::Match
  - Will be removed in **27.6**
- EBarCodeReader::EnablePermissiveDecoding
  - Replaced by EnablePatternCorrection, EnableAdvancedDecoding, EnablePrintingErrorCorrection
  - Will be removed in **27.6**
- EChecker2 (in EasyObject)
  - Replaced by the new EChecker class (in EasyObject)
  - Will be removed in **27.6**
- EMatrixCodeReader, EBarCodeReader, EQRCodeReader, ECodeReader Read overloads without EGrid parameter
  - Replaced by overloads taking EGrid
  - Will be removed in **27.6**

- ETextReader::Localize
    - Replaced by ETextReader::Locate
    - Will be removed in **27.6**

*Deprecated in **26.02***

- SetImagePtr for classes EBaseROI, EFourierImage, EZMap and EDepthMap
    - Replaced by method UseExternalBuffer
    - Will be removed in **27.6**
- EPoint constructor taking EFrame as argument
    - Replaced by other EPoint constructors
    - Will be removed in **27.6**
- EFlipAxis enum values
    - Replaced by new enum values following naming convention (for ex. EFlipAxis_Horizontal)
    - Will be removed in **27.6**
- E3DPlane::FlipIfNegativeZ
    - Replaced by E3DPlane::OrientNormal(E3DPlaneNormalOrientation_PosZ)
    - Will be removed in **27.6**

*Deprecated in **25.10***

- EPoint::SetCenter, EPoint::GetCenter, EPoint::SetCenterXY
    - Replaced by direct access to X and Y properties and other EPoint methods
    - Will be removed in **27.6**

*Deprecated in **25.06***

- EPointCloudMerger::Merge (non-pointer-vector overload)
    - Replaced by EPointCloudMerger::Merge overload taking std::vector<EPointCloud*>
    - Will be removed in **27.6**

*Deprecated in **25.02***

- EPatternFinder::Learn with don’t care area mask
    - Replaced by the use of ERegion in EPatternFinder::Learn
    - Will be removed in **27.6**
- EMatcher::DontCareThreshold
    - Replaced by the use of ERegion in EMatcher::LearnPattern
    - Will be removed in **27.6**
- EMatcher::GetNumPositions, EMatcher::GetPosition, EMatcher::GetPositions

  - Replaced by the vector of EFoundPattern returned by EMatcher::Match
  - Will be removed in **27.6**
- EMatchPosition struct for EMatcher
  - Replaced by EFoundPattern
  - Will be removed in **27.6**
- EBarCodeReader::EnableAllSymbologies
  - Replaced by EBarCodeReader::EnableSafeSymbologies
  - Will be removed in **27.6**
- ESerializer::CreateFileWriter, CreateFileReader, CreateMemoryWriter, CreateMemoryReader, GetWriting
  - Replaced by new EFileSerializer and EMemorySerializer constructors
  - Will be removed in **27.6**

*Deprecated in **24.10***

- EImageEncoder::Serialize
  - Replaced by EImageEncoder::Save or EImageEncoder::Load
  - Will be removed in **27.6**
- Serialize for EPen, EFont, EBrush classes
  - Replaced by methods Save or Load
  - Will be removed in **27.6**
- EBarCodeReader:SetMinModuleSize, GetMinModuleSize, GetUseMinModuleSize, SetUseMinModuleSize
  - Will be removed in **27.6**

*Deprecated in **24.06***

- Easy::Initialize and Easy::Terminate
  - No replacement, it is now done automatically
  - Will be removed in **27.6**
- EMatcher::GetExtension(int32_t&,int32_t&) and SetExtension(int32_t,int32_t)
  - Replaced by EMatcher::Set/GetExtension with EIntSize type
  - Will be removed in **27.6**
- EMatcher::GetPixelDimensions(float&,float&) and EMatcher::SetPixelDimensions(float,float)
  - Replaced by EMatcher::Set/GetPixelDimensions with ESize type
  - Will be removed in **27.6**
- E3DMatcher::GetAnomalyThresholds(float&,float&) and E3DMatcher::SetAnomalyThresholds(float,float)
  - Replaced by E3DMatcher::Set/GetAnomalyThresholds with EAnomalyThresholds type

  - Will be removed in **27.6**

- E3DComparer::GetAnomalyThresholds(float&,float&) and E3DComparer::SetAnomalyThresholds(float,float)
  - Replaced by E3DComparer::Set/GetAnomalyThresholds with EAnomalyThresholds type
  - Will be removed in **27.6**
- E3DMatcher::GetAnomalyHysteresis(float&,float&) and E3DMatcher::SetAnomalyHysteresis(float,float)
  - Replaced by E3DMatcher::Set/GetAnomalyHysteresis with EAnomalyThresholds type
  - Will be removed in **27.6**
- E3DComparer::GetAnomalyHysteresis(float&,float&) and E3DComparer::SetAnomalyHysteresis(float,float)
  - Replaced by E3DComparer::Set/GetAnomalyHysteresis with EAnomalyThresholds type
  - Will be removed in **27.6**
- E3DMatcher::GetEdgeCroppingParameters(int&,float&) and E3DMatcher::SetEdgeCroppingParameters(int,float)
  - Replaced by E3DMatcher::Set/GetEdgeCroppingParameters with EEdgeCroppingParameters type
  - Will be removed in **27.6**
- E3DComparer::GetEdgeCroppingParameters(int&,float&) and E3DComparer::SetEdgeCroppingParameters(int,float)
  - Replaced by E3DComparer::Set/GetEdgeCroppingParameters with EEdgeCroppingParameters type
  - Will be removed in **27.6**
- E3DTransformMatrix::GetAzimuthElevationAngles(float&,float&)
  - Replaced by EAzimuthElevationAngles E3DTransformMatrix::GetAzimuthElevationAngles
  - Will be removed in **27.6**
- E3DViewer::GetColorRampLocation(float&,float&,float&,float&) and E3DViewer::SetColorRampLocation(float,float,float,float)
  - Replaced by E3DViewer::Set/GetColorRampLocation with EPoint parameters
  - Will be removed in **27.6**
- E3DViewer::GetFixColorRampBounds(float&,float&) and E3DViewer::SetFixColorRampBounds(float,float)
  - Replaced by E3DViewer::Set/GetFixColorRampBounds with EFloatRange type
  - Will be removed in **27.6**
- E3DViewer::GetViewTarget(float&,float&,float&) and E3DViewer::SetViewTarget(float,float,float)
  - Replaced by E3DViewer::Set/GetViewTarget with E3DPoint type
  - Will be removed in **27.6**
- E3DViewer::GetViewAngle(float&,float&,float&) and E3DViewer::SetViewAngle(float,float,float)
  - Replaced by E3DViewer::Set/GetViewAngle with E3DPoint type
  - Will be removed in **27.6**

- E3DViewer::GetAutoRotate(float&,float&,float&) and E3DViewer::SetAutoRotate(float,float,float)
    - Replaced by E3DViewer::Set/GetAutoRotate with E3DPoint type
    - Will be removed in **27.6**
- ConvertCoordinatesMapToPixel(float,float,int&,int&) for EZMap and EDepthMap classes
    - Replaced by ConvertCoordinatesMapToPixel(EPoint,EIntPoint&)
    - Will be removed in **27.6**
- ConvertCoordinatesPixelToMap(int,int,float&,float&) for EZMap and EDepthMap classes
    - Replaced by EPoint ConvertCoordinatesPixelToMap(EIntPoint)
    - Will be removed in **27.6**
- EZMap::GetSizeInWorld, GetResolution, ImageToZMap, ZMapToImage, GetPixelPositionFromWorldPosition (ref-based overloads)
    - Replaced by overloads returning/using EPoint, E3DPoint, EIntPoint
    - Will be removed in **27.6**
- E3DPlane EPlaneFinder::Find(EPointCloud,...)
    - Replaced by float EPlaneFinder::Find(EPointCloud,E3DPlane&,...)
    - Will be removed in **27.6**
- E3DPlane EPlaneFitter::Fit(EPointCloud,...)
    - Replaced by float EPlaneFitter::Fit(EPointCloud,E3DPlane&)
    - Will be removed in **27.6**
- ESphereFitter::Fit(...) returning E3DSphere
    - Replaced by version returning ratio of inliers with sphere as argument
    - Will be removed in **27.6**
- EPrincipalAxisExtractor::Extract (ref-based overload)
    - Replaced by overload using E3DPoint
    - Will be removed in **27.6**
- EStatistics::ComputeStatistics, EStatistics::ComputePixelStatistics (ref-based overloads)
    - Replaced by overloads returning EStatistics8, EStatistics16 or EStatistics32f
    - Will be removed in **27.6**
- EPointCloud::AllocateAttributeBuffer, FillAttributeBuffer, AddCustomAttributeBuffer, AllocateCustomAttributeBuffer, SetAttribute (generic overloads)
    - Replaced by named-type method variants
    - Will be removed in **27.6**
- EasyImage::HistogramThreshold and EasyImage::HistogramThresholdBW16 (ref-based overloads)

  - Replaced by overloads returning EHistogramThreshold
  - Will be removed in **27.6**
- EasyImage::TwoLevelsMinResidueThreshold (ref-based overload)
  - Replaced by overload returning ETwoLevelsMinResidueThreshold
  - Will be removed in **27.6**
- EasyImage::ThreeLevelsMinResidueThreshold (ref-based overload)
  - Replaced by overload returning EThreeLevelsMinResidueThreshold
  - Will be removed in **27.6**
- EasyImage::Thin, EasyImage::BiLevelThin, EasyImage::Thick, EasyImage::BiLevelThick (void overloads)
  - Replaced by overloads returning the number of iterations performed
  - Will be removed in **27.6**
- EasyImage::MatchFrames (all overloads)
  - Replaced by overloads returning the optimal shift amplitude
  - Will be removed in **27.6**
- EKernel::GetKernelData (ref-based overload)
  - Replaced by overload returning the coefficient value
  - Will be removed in **27.6**
- EMovingAverage::GetSize (ref-based overload)
  - Replaced by individual size getters
  - Will be removed in **27.6**
- EasyImage::PixelAverage, PixelStat, PixelStatBW8, PixelStatBW16, PixelStdDev, PixelCount, Area, AreaDoubleThreshold, PixelVariance, GravityCenter, BinaryMoments, WeightedMoments
  - Replaced by EPixelStatisticsBW8, EPixelStatisticsBW16, EPixelStatisticsC24 classes
  - Will be removed in **27.6**
- EClassificationDataset::GetImages returning vector<EBaseROI*>
  - Replaced by EClassificationDataset::GetImageCopy
  - Will be removed in **27.6**
- EasyObject::ContourArea(EPathVector*, int32_t&)
  - Replaced by int32_t EasyObject::ContourArea(EPathVector*)
  - Will be removed in **27.6**
- EasyObject::ContourGravityCenter, EasyObject::ContourInertia
  - Replaced by EasyObject::ContourStatistics
  - Will be removed in **27.6**

- EImageEncoder::GetBinaryImageSegmenter and Segmenters::EBinaryImageSegmenter
    - Will be removed in **27.6**
- ESelectByPosition, ESelectOption, ESortOption enums
    - Replaced by ECodedImage2
    - Will be removed in **27.6**
- SetMinNumFitSamples(int32_t, int32_t) for ECircleGauge, ELineGauge, ERectangleGauge, EWedgeGauge
    - Replaced by overloads using EIntegerRange
    - Will be removed in **27.6**
- AddSkipRange and GetSkipRange (int32_t overloads)
  for ECircleGauge, ELineGauge, ERectangleGauge, EWedgeGauge
    - Replaced by overloads using EIntegerRange
    - Will be removed in **27.6**
- EVectorModel::GetVectorModelExtremas (old overload)
    - Replaced by overloads using EFloatRange
    - Will be removed in **27.6**
- EasyQRCodeReader::SetSearchField
    - Replaced by EasyQRCodeReader::Read
    - Will be removed in **27.6**

*Deprecated in **24.02.1***

- EPlaneFinder::GetSeed and EPlaneFinder::SetSeed
    - Will be removed in **27.6**

*Deprecated in **24.02***

- E3DAnomaly::Create3DObject
    - Replaced by E3DAnomaly::GetBoundingBox
    - Will be removed in **27.6**
- OverallSymbolGrade field in EMatrixCodeIso15415GradingParameters,
  EMatrixCodeIso29158GradingParameters, EQRCodeIso15415GradingParameters
    - Replaced by ScanGrade
    - Will be removed in **27.6**

*Deprecated in **23.12***

- EGDIPlusDrawAdapter
    - Replaced by EWindowsDrawAdapter
    - Will be removed in **27.6**

- Easy::OpenImageGraphicContext and Easy::CloseImageGraphicContext
    - Will be removed in **27.6**
- EImageBW1
    - Will be removed in **27.6**
- ECodedImage, EFeatureData, EListItem
    - Replaced by ECodedImage2
    - Will be removed in **27.6**
- EBarCode (EasyBarCode1)
    - Replaced by Euresys::Open_eVision::EasyBarCode2::EBarCodeReader
    - Will be removed in **27.6**
- EMatrixCode (EasyMatrixCode1)
    - Replaced by Euresys::Open_eVision::EasyMatrixCode2::EMatrixCode
    - Will be removed in **27.6**
- EMatrixCodeReader (EasyMatrixCode1)
    - Replaced by Euresys::Open_eVision::EasyMatrixCode2::EMatrixCodeReader
    - Will be removed in **27.6**
- ESearchParamsType
    - Will be removed in **27.6**
- EasyColor::C24ToBayer(EROIC24*, EROIBW8*, bool evenCol, bool evenRow)
    - Replaced by overload taking EBayerConfiguration
    - Will be removed in **27.6**
- EasyColor::TransformBayer(EROIBW8*, EROIBW8*, EColorLookup*, bool evenCol, bool evenRow)
    - Replaced by overload taking EBayerConfiguration
    - Will be removed in **27.6**
- Draw methods with HDC API
    - Replaced by the overload taking an EDrawAdapter by using an instance of EWindowsDrawAdapter
    - Will be removed in **27.6**

*Deprecated in **23.08***

- EOCR::ReadTextWide
    - Replaced by EOCR::ReadText
    - Will be removed in **27.6**
- EOCR::RecognizeWide

- Replaced by EOCR::Recognize
- Will be removed in **27.6**

*Deprecated in **23.04***

- EDeepLearningTool::SetEnableGPU and EDeepLearningTool::GetEnableGPU
  - Will be removed in **27.6**
- EDeepLearningTool::GetNumGPUs
  - Replaced by EDeepLearningTool::NumAvailableDevices
  - Will be removed in **27.6**
- EDeepLearningTool::SetGPUIndexes and EDeepLearningTool::GetGPUIndexes
  - Replaced by EDeepLearningTool::ActiveDevices
  - Will be removed in **27.6**
- EQuadrangle::IsInside()
  - Replaced by EQuadrangle::IsPointInside
  - Will be removed in **27.6**
- EBarCodeReader::GetValidateWithChecksum and EBarCodeReader::SetValidateWithChecksum
  - Replaced by Set/GetValidateMandatoryChecksum, Set/GetValidateOptionalChecksum, Set/GetValidatePharmacode
  - Will be removed in **27.6**
- EBarCodeReader::GetValidateBarCode and EBarCodeReader::SetValidateBarCode
  - Replaced by Set/GetValidateMandatoryChecksum, Set/GetValidateOptionalChecksum, Set/GetValidatePharmacode
  - Will be removed in **27.6**

*Deprecated in **22.12***

- EClassifier::SetCapacity
  - Replaced by EClassifier::SetModelType
  - Will be removed in **27.6**

*Deprecated in **22.08***

- ELocatorMetrics::GetBestWeightedFScoreAndThreshold
  - Replaced by ELocatorMetrics::GetBestWeightedFScorePoint
  - Will be removed in **27.6**
- ELocatorMetrics::GetBestWeightedPrecisionAndThreshold
  - Replaced by ELocatorMetrics::GetBestWeightedPrecisionPoint
  - Will be removed in **27.6**

- ELocatorMetrics::GetBestWeightedRecallAndThreshold
    - Replaced by ELocatorMetrics::GetBestWeightedRecallPoint
    - Will be removed in **27.6**
- ELocatorMetrics::GetLabelIntersectionOverUnion
    - Replaced by ELocatorMetrics::GetLabelAverageProximity
    - Will be removed in **27.6**
- ELocatorMetrics::GetIntersectionOverUnion
    - Replaced by ELocatorMetrics::AverageProximity
    - Will be removed in **27.6**

*Deprecated in **2.16***

- EWorldShape::SetDistortion
    - Replaced by EWorldShape::SetDistortionStrength
    - Will be removed in **27.6**

*Deprecated in **2.15***

- EChecker (EasyOCV)
    - Replaced by EChecker2
    - Will be removed in **27.6**

*Deprecated in **2.13***

- EasyQRCode::Detect and EasyQRCode::Decode
    - Replaced by EasyQRCode::Read
    - Will be removed in **27.6**

*Deprecated in **2.11***

- EasyOCV library (EOCV class, ECharCreationMode, EDegreesOfFreedom, EDiagnostic, ELearningMode, ELocationMode, EQualityIndicator enums)
    - Will be removed in **27.6**

CONTACT US

Website, email

GENERAL

www.alliedvision.com/en/contact
info@alliedvision.com

DISTRIBUTION PARTNERS

www.alliedvision.com/en/avt-locations/avt-distributors

SUPPORT

www.alliedvision.com/en/support
www.alliedvision.com/en/about-us/contact-us/technical-support-repair-/-rma

Offices

EUROPE, MIDDLE EAST, AND AFRICA (HEADQUARTERS)

Allied Vision Technologies GmbH
Taschenweg 2a
07646 Stadtroda, Germany
T// +49 36428 677-0 (Reception)
T// +49 36428 677-230 (Sales)
F// +49 36428 677-28

ASIA-PACIFIC CHINA

Allied Vision Technologies Shanghai Co Ltd.
B-510, Venture International Business Park
2679 Hechuan Road
Minhang District, Shanghai 201103
People's Republic of China
T// +86 21 64861133

SINGAPORE

Allied Vision Technologies Asia Pte. Ltd
82 Playfair Rd, #07-01 D'Lithium
Singapore 368001
T// +65 6634 9027

NORTH, CENTRAL, AND SOUTH AMERICA, CANADA

Allied Vision Technologies Canada Inc.
300 – 4621 Canada Way
Burnaby, BC V5G 4X8, Canada
T// +1 604 875 8855

USA

Allied Vision Technologies, Inc.
102 Pickering Way - Suite 502
Exton, PA 19341, USA
Toll-free// +1-877-USA-1394
T// +1 978 225 2030

JAPAN

Allied Vision Technologies
Yokohama Portside Bldg. 10F
8-1 Sakae-cho, Kanagawa-ku
Yokohama-shi, Kanagawa, 221-0052
T// +81 (0) 45 577 9527

Liability, trademarks, and copyright

Allied Vision has tested the product under the described conditions. The customer assumes all risk of product damage, application compromise or potential failure, and Sales Warranty loss related to deviation from the described conditions. Allied Vision's acknowledgement of such deviations in the customer's modified product or applications does not constitute advice for use. No Warranty is offered or implied by Allied Vision regarding the customer's assumed risk or legal responsibilities with such modified products or applications. All text, pictures, and graphics are protected by copyright and other laws protecting intellectual property. All content is subject to change without notice. All trademarks, logos, and brands cited in this document are property and/or copyright material of their respective owners. Use of these trademarks, logos, and brands does not imply endorsement.
Copyright © 2026 Allied Vision Technologies GmbH. All rights reserved.