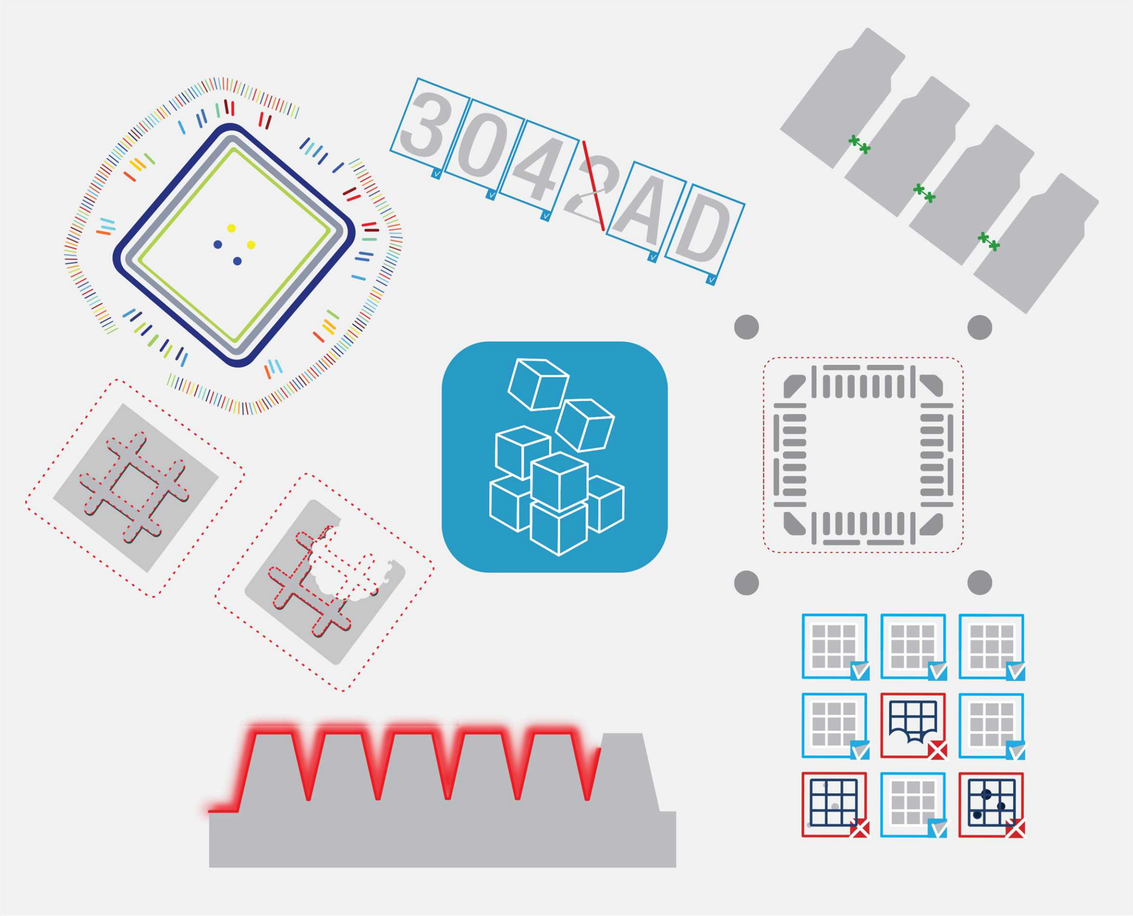
RELEASE NOTES

Open eVision

Release 2.11.1



Terms of Use

This documentation is provided with Open eVision 2.11.1 (doc build 1125).

Contents

1. Release Summary

Release benefits

Deep learning tools

- The deep learning tools are based on deep convolutional neural networks (CNNs):
  - □ **EasyClassify** classifies images into a predefined set of classes.
    - Use it to identify a product in an image.
    - Use it to detect if a product is good or defective.
  - □ **EasySegment** detects and segments defects in images.
    - It works in an unsupervised way.
    - Train it using "good" images.
    - Use it to segment and classify new images as good or not.
- The deep learning tools now support regions of interest (ROI) and masks.
- **EasyClassify** and **EasySegment** are only available in the deep learning bundle and cannot be purchased separately.

EasyGauge

- **EasyGauge** now supports `ERegion` on all gauges when you call `Measure()`.

EasyQRCode

- The detection and the speed of the `Read` method are improved.
- The `GetDecodedString` method returns the information contained in a QR code as a concatenated utf-8 string.

Installers

- Each Studio applications of Open eVision has its own installer.

  As this change required a rework of the installer and to avoid any upgrade issue, uninstall any previous version of Open eVision 2.11 before installing this new version.

Deprecated items

EasyOCV

- The **EasyOCV** tool is deprecated:
    - □ Starting with Open eVision 2.11, the corresponding sample programs and documentation are removed.
    - □ This tool is also removed from Open eVision Studio.
- Nevertheless, the corresponding functions are still available in the libraries.

EasyFind

- The contrasting regions pattern type is deprecated.
    - □ Starting with Open eVision 2.11, the corresponding sample programs and documentation are removed.
    - □ This option is also removed from Open eVision Studio.
- Nevertheless, the corresponding option is still available in the libraries.

2. Release Specifications

OS and processor architectures

- Open eVision is a 32-bit and 64-bit library that requires a processor compatible with the SSE2 instruction set.
- Open eVision runs on the following operating systems:

OS version	Additional information	
Windows 10®	32-bit	—
Windows 10®	64-bit	—
Windows 8®	32-bit	—
Windows 8®	64-bit	—
Windows 7®	32-bit	The recommended version is 6.1.7601 (Windows 7 Service Pack 1)
Windows 7®	64-bit	

- Remote connections:
  - □ You can install and use Open eVision licenses on a remote connection using remote desktop, TeamViewer or any other similar software.
- Virtual machines:
  - □ You cannot install Open eVision on virtual machines.

Supported IDE and programming languages

Select the recommended API Module according to your IDE and programming language:

	Programming language	
IDE	C++	C#, VB.NET, C++/CLI
Microsoft Visual Studio 2008® SP1	C++	.NET Assembly
Microsoft Visual Studio 2010®	C++	.NET Assembly
Microsoft Visual Studio 2012®	C++	.NET Assembly
Microsoft Visual Studio 2013® (*)	C++	.NET Assembly
Microsoft Visual Studio 2015®	C++	.NET Assembly
Microsoft Visual Studio 2017®	C++	.NET Assembly

NOTE
(*) Visual C++ MFC MBCS Library for Visual Studio 2013 must be installed.

Required system resources

- Display size:
    - Minimum: 800 x 600
    - Recommended: 1280 x 1024
- Display color depth:
    - Minimum: 16 bits
    - Recommended: 32 bits
- Hard disk space:
    - Open eVision libraries: 100 MB - 2 GB (depending on selected options)

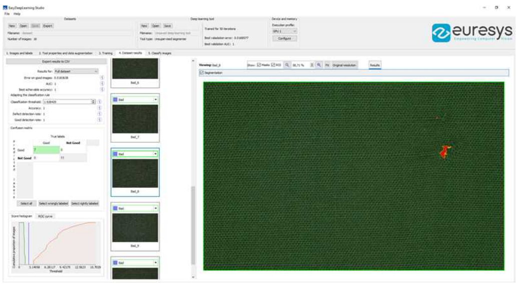
3. Release Details

3.1. New and Improved Features

New features

Deep learning tools

- The deep learning tools (previously named **EasyDeepLearning**) are based on deep convolutional neural networks (CNNs):
  - □ **EasyClassify** classifies images into a predefined set of classes.
    - Use it to identify a product in an image.
    - Use it to detect if a product is good or defective.
  - □ **EasySegment** detects and segments defects in images.
    - It works in an unsupervised way.
    - Train it using "good" images.
    - Use it to segment and classify new images as good or not.



- The deep learning tools now support regions of interest (ROI) and masks.

EasyGauge

- **EasyGauge** now supports `ERegion` on all gauges when you call `Measure()`.

EasyQRCode

- The `EQRCodeReader::Detect` and `EQRCodeReader::Read` methods now return `QRCode` objects and geometries vectors in lines then columns order from the top left corner of the image, based on their gravity center position.

- **EasyQRCode** supports Extended Channel Interpretation (ECI) with ECI table Indicator 26 (UTF-8).
  - □ Data from QR codes with other ECI table indicators are available in byte form or as a string in the byte part converted to hexadecimal.
  - □ Use the `GetECITableIndicator` method of the `EQRCodeDecodedStreamPart` object to access the ECI table indicator.
- The `EQRCode` and `EQRCodeDecodedStreamPart` objects have a `GetDecodedString` method that can take an `EByteInterpretation` enumeration as parameter.
- The `GetDecodedString` method returns the information contained in a QR code as a concatenated utf-8 string. To interpret the non UTF8-ECI codes of byte parts, use an input in the form of an `EByteInterpretation` parameter.
- C++ code sample now allows the selection of the `EQRDetectionTradeOff` and the selection of the `EByteInterpretationMode` for the decoded string.
- C++ code sample now supports the display of the Kanji encoding.

EasyFind

- The new added `GetFeaturePoints` method of the `EPatternFinder` class returns the information about the feature points of the learned pattern.

Improvements

EasyClassify

- New variants of the metrics in `EClassificationMetrics` are added: weighted (with respect to the label weights), balanced and per-label metrics

EasyClassify and EasySegment

- The new `EClassificationDataset::GetImageCopyWithDataAugmentation` method is available.

EasyQRCode

- The detection and the speed of the `Read` method are improved.
- The reference documentation is improved and additional links ease the navigation.

EasyOCR2

- The detected text has a more accurate angle in some edge cases such as texts containing commas or apostrophes.

EasyMatch

- The new `EError_InsufficientDiscriminantFeaturesInPattern` exception is raised when you try to learn a pattern with not enough discriminant features (only in Advanced Learning mode).

Easy3D Studio

- **Easy3D Studio** now exposes the Z-resolution parameter to the depth map simulation parameters. Thanks to this parameter, you can use external images as depth maps.

Easy3DObject

- `E3DObject` proposes the new `GetAspectRatio` query.

3.2. Deprecated Items

Starting with this release 2.11, Open eVision does not support the following items:

EasyOCV

- The **EasyOCV** tool is deprecated:
  - □ Starting with Open eVision 2.11, the corresponding sample programs and documentation are removed.
  - □ This tool is also removed from Open eVision Studio.
- Nevertheless, the corresponding functions are still available in the libraries.

EasyFind

- The contrasting regions pattern type is deprecated.
  - □ Starting with Open eVision 2.11, the corresponding sample programs and documentation are removed.
  - □ This option is also removed from Open eVision Studio.
- Nevertheless, the corresponding option is still available in the libraries.

3.3. Breaking Changes

Starting with this release 2.11, Open eVision implements the following changes:

Multicore processing

- `Easy::SetMaxNumberOfProcessingThreads` sets the number of Open eVision processing threads **only** for the thread in which it is called.

EasyClassify and EasySegment

- Replace the function `EClassificationDataset::GetImage` by `EClassificationDataset::GetImageCopy`.
- `GetBatchSizeForMaximumClassificationSpeed` is renamed `GetBatchSizeForMaximumInferenceSpeed`.

EasyQRCode

- The `GetPosition` method of the `EQRCodeGeometry` object now returns an `EQuadrangle` object instead of an `EQuadrilateral` object.

3.4. Changes

Starting with this release 2.11, Open eVision implements the following changes:

Multithreading

- In a multithread application, call `Easy::Initialize` in your main thread before starting any other thread that performs processing with Open eVision.

EasyClassify and EasySegment

- The computation of the classification metrics during the training is changed to be compatible with the `EClassifier::Evaluate` method.
- The `EClassifier` class is now a child of the `EDeepLearningTool` class.

Installers

- Each Studio applications of Open eVision has its own installer.

> ❗ **WARNING**
> As this change required a rework of the installer and to avoid any upgrade issue, uninstall any previous version of Open eVision 2.11 before installing this new version.

3.5. Solved Issues

The following issues have been fixed in Open eVision 2.11:

EasysMatrixCode2

- If `MaxNumCodes` is set to '`0`', the reader now tries to find as many `EMatrixCode` as possible.
- **EasyMatrixCode2** now detects and decodes non-square matrix codes.

Multicore processing

- The default behavior (multicore processing disabled) is now correctly set in new threads and these new threads do not use all the cores available on the machine.

EasyQRCode

- The `GetIsDecodingReliable` value that could be wrong in the case of a false-positive QR code reading is now correct in all cases.
- The decoding of a badly detected QR code does not crash the program anymore.
- Over-sized QR codes could be rendered using finder patterns of multiple QR codes.
- The `EQRDetectionMethod_PerspectiveLegacy` mode is fixed.

Easy3D

- The conversion from `EDepthMap32f` to `EPointCloud`, using the `EDepthMapToPointCloudConverter` class, is now correctly handled.
- **Easy3D Studio** is now compatible with **Coaxlink** firmware 11.2 and above.
- In **Easy3D Studio**, the default profiles and depth maps folders are now correct and you can edit them.

EasySegment

- The training of an unsupervised segmentation without any defective image does not fail anymore.
- The training with the CPU does not fail anymore.

EasyOCR2

- The small error in the angle determination when the angle range is 360 degrees, that appeared with the recent improvements, is now fixed.

Open eVision Studio

- In the visualization of the EOCR segmentation, the pixels with a value equal to the threshold are now correctly marked.
- When using **EasyGauge**, the tolerance value in the wedge gauge parameter panel is now correctly updated when you edit the graphical representation.
- With **EasyGauge**, when you add a wedge gauge on a calibrated world shape, Open eVision Studio does not crash anymore.

Deep Learning Studio

- When you open images in the Classify images tab, **Deep Learning Studio** does not crash anymore.
- When you refresh a result or access the heat map of an image, **Deep Learning Studio** does not crash anymore as it did sometimes.

General

- The `EBaseROI` frame is correctly drawn (with `EBaseROI::DrawFrame` method) when using non null values for `panX` and `panY` parameters.

Regions

- The union of a region with an empty region now has a correct bounding box.

4. Known Issues

.NET API and unsigned integer parameters

Since this release 2.5 of Open eVision, unsigned integer parameters in the C++ API are not exposed in the .NET API as signed integer anymore, but as unsigned integers. This brings the .NET API closer to the C++ one.

This change does not cause any issue except when you want to pass an enumerate value as one of these parameters. In these specific cases, update your casting operation as in the following example:

```
codedImage.SetThreshold((int)EThresholdMode.MinResidue);
```

becomes:

```
codedImage.SetThreshold(unchecked((uint)EThresholdMode.MinResidue)); .
```

Reserved keywords

The following keywords are reserved by Open eVision:

- EUnit_um, EUnit_mm, EUnit_cm, EUnit_dm
- EUnit_m, EUnit_dam, EUnit_hm, EUnit_km
- EUnit_mil, EUnit_inch, EUnit_foot, EUnit_yard, EUnit_mile
- EasyWorld

TIP
To avoid conflict, do not use these keywords to name variables, functions, methods, macros...

Image formats

- If you use some types of 96-bit RGB Tiff image, Open eVision may crash.

Object cleanup: .NET

As a rule, it is highly recommended to call `Dispose()` on Open eVision .NET objects when they are not useful anymore.

TIP
Not doing so might result in unnecessarily high memory usage and crashes.

Example in C#

```
using(EImageBW8 src = new EImageBW8())
using(EPatternFinder finder = new EPatternFinder())
{
  src.Load(ImageFilePath);
  EFoundPattern[] foundPatterns = finder.Find(src);
  ...
  foreach(EFoundPattern foundPattern in foundPatterns)
  {
    foundPattern.Dispose();
  }
}
```

In addition, if you use a nested object (such as the segmenter properties in EasyObject encoder objects), remember to call `Dispose()` on that object before calling `Dispose()` on the parent object.

Example in C#

```
imageEncoder.GrayscaleSingleThresholdSegmenter.BlackLayerEncoded = true;
...
imageEncoder.GrayscaleSingleThresholdSegmenter.Dispose();
imageEncoder.Dispose();
```

Basic types: retrieving and setting pixel values

Using the `GetPixel()` and `SetPixel()` methods of the various ROI classes can sometimes be slow if you make many calls (regardless of the language used).

- In order to greatly speed up the ROI/image buffer access, embed the buffer access in your own code.
- See the examples below that use the new Open eVision API.

NOTE
For a better readability of these examples, the variable declarations and initializations have been omitted when possible.

Example in C++

```
void* pixAddr;
UINT8 pix;
...
for (int y = 0; y < height; ++y)
{
  pixAddr = bw8Image.GetImagePtr(0,y);
  for (int x = 0; x < width; ++x)
  {
    pix = *(reinterpret_cast<UINT8*>(pixAddr)+x);
  }
}
```

Example in C#

```
using System.RunTime.InteropServices;
...
IntPtr pixAddr;
byte pix;
...
for (int y = 0; y < height; ++y)
{
  pixAddr = bw8Image.GetImagePtr(0,y)
  for (int x = 0; x < width; ++x)
  {
    pix = Marshal.ReadByte(pixAddr,x)
  }
}
```

Basic types: ROI zooming and panning issue

- When drawing an ROI with a zoom factor, applying panning (retrieved from a scroll bar) causes the ROI display to be shifted. Consequently, the `HitTest()` and `Drag()` functions fail because the handles do not appear at their actual positions.

  Workaround: The panning values should be divided by the zoom factor before calling the `DrawFrame()`, `HitTest()` and `Drag()` functions.

Basic Types: miscellaneous issues

- TIFF files containing RGB values + alpha values are not supported.
- Filenames with multibyte characters are not supported. The error is "Unrecognized file format".
- `Easy::GetBestMatchingImageType()` only works for BW8 and C24 images.

Multithreading

- In multithread applications, if `Easy::Initialize` is not called before launching new threads that call Open eVision functions, then the number of Open eVision processing threads in these new threads may be wrongly initialized to use all the cores that are available on the machine.

EasyBarCode

- Due to a bug in the debugger of Visual C++ 2012, the reading time of bar codes may increase after a failed reading. This happens only in debug mode with Visual C++ 2012.
- EasyBarCode requires that a quiet zone of at least one full module is present around the whole bar code to be read.
- EasyBarCode is currently unable to read bar codes with curved or distorted bars. For reliable reading, the bars must be as straight as possible.
- EasyBarCode is currently not multithread-safe.

EasyQRCode

- EasyQRCode does not support MicroQR codes.

EasyObject

- The ECodedImage2 and EHarrisDetector results are drawn slowly when there are many results.

EasyMatch

- By design, the maximum size for a pattern in EasyMatch is 1791 x 1791.
- 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

- 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 Open eVision Studio or in Open eVision Eval.

  Workaround: Enter numerical values for the wedge gauge position.

EasyMatrixCode

- 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.
- In .NET, retrieving the coordinates of a MatrixCode using EMatrixCode.GetCorner() or EMatrixCode.Center() can lead to an unhandled exception when the garbage collection starts up. To avoid this problem, call Dispose() on the EPoint objects returned by these functions when they are no longer needed.

Open eVision Studio

- In the ROI management dialog, clicking on a ROI in the tree view does not activate the ROI overlay in the image window. This can prevent you to graphically interact with it.

  To avoid this issue and to properly interact with the ROI overlay:

  a. Click on the ROI in the tree view.

  b. Immediately after, click inside its overlay in the image window.
- To avoid crashes, deselecting all detection methods in the EasyQRCode dialog box reverts to the default detection method. In some cases, the dialog might not refresh automatically.
- In the detection method selection control of the EasyQRCode dialog box, clicking beside a text might select or deselect it.
- When managing the EasyOCR2 topology, the potential characters option is not available.

Open eVision installer

- There is a conflict between the Open 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 Open eVision first, and then the other software.

> **NOTE**
> This port is typically used by National Instrument software such as LabView.

- Before installing any Euresys product, make sure that your OS is up-to-date (using Microsoft Update), otherwise, problems might occur.

Open eVision License Manager

- Under Windows XP, the Open eVision License Manager might not start if the .NET Framework 2.0 is not installed.
- Using the Open eVision License Manager to activate a license requires an Internet connection and a secure SSL transaction to EURESYS s.a. servers.

> **NOTE**
> On older systems, such as Windows XP SP3, ensure that the root certificates are up-to-date otherwise the secure connection is refused and the license is not activated.

- When activating an emergency license, the following error may occur: “`Error Message: Loading of the ASR failed!`”

  This error occurs when all 3 emergency licenses have already been used and the computer has been formatted.

- Using Open eVision License Manager 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 Open eVision License Manager runs in the OS language, including Chinese and Japanese.