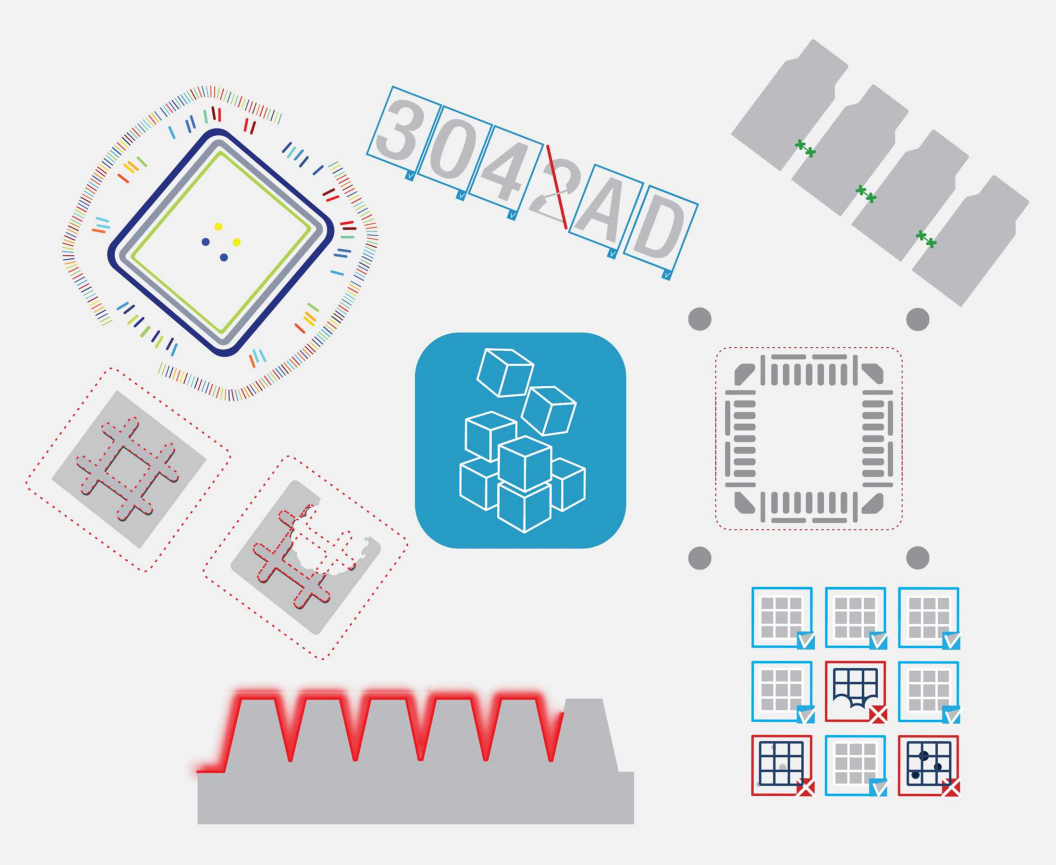


Open eVision

Checking Lot Codes and Mixing with OCR



This documentation is provided with **Open eVision 24.02.0** (doc build **1198**).
www.euresys.com

This documentation is subject to the General Terms and Conditions stated on the website of **EURESYS S.A.** and available on the webpage <https://www.euresys.com/en/Menu-Legal/Terms-conditions>. The article 10 (Limitations of Liability and Disclaimers) and article 12 (Intellectual Property Rights) are more specifically applicable.

1. Checking Lot Codes and Mixing with OCR

The code of this application and the images used are available in your public document folder (C:\Users\Public\Documents\Euresys\Open eVision xx):

Application objective

This application demonstrates how to check the lot codes of components, for example, during the quality control phase in the semiconductor final packaging process. It uses EasyOCR to check that only components from a specific lot are present in the final packaging.

The application is divided in the following 3 programs that illustrate different parts of the process:

1. Learning lot mixing
 - This program learns the characters used in the checking of the lot mixing process.
 - You can save the resulting font file.



NOTE

To run this program, you need the EasyOCR license.

2. Exploring OCR font
 - The EasyOCR character recognition relies on the usage of proprietary font files.
 - A font file stores the segmentation parameters allowing to extract characters from the inspected image, as well as the set of learned characters.
 - Each character is stored as a small bitmap, called a pattern.
 - This program shows how to browse the list of patterns stored in an EasyOCR font file.
 - It can be used as a font file maintenance tool.



NOTE

To run this program, you need the EasyOCR license.

3. Checking of lot mixing online
 - This program exhibits the final inspection using EasyOCR.
 - It reads and reports the characters of each device.



NOTE

To run this program, you need the EasyOCR, EasyImage and EasyGauge licenses.

Learning lot mixing

Initializing

1. During the learning process, the application needs an image featuring the set of characters used on the device.

**TIP**

In this example it uses the OCR-B.bmp image that contains all white uppercase and numeric OCR-B characters on a black background.

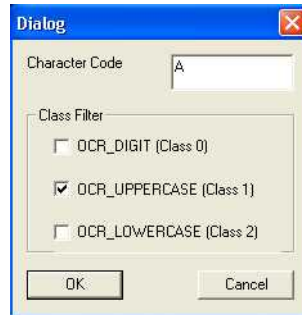
2. The application segments this image to find the character blobs.
3. It draws a red bounding box around each character and extracts the characters.



Learning

Using the CharCodeDialog dialog box that is invoked whenever you double-click on a character in the image and that is illustrated below:

1. Associate each character with the corresponding code.
2. For each character, also set its class.



EasyOCR adds the character pattern, the corresponding code and the class into its memory.

Saving

1. Open the **File** menu and click on **Save**.
2. Save the learned characters into an EasyOCR font file to use it later.



Exploring the OCR font

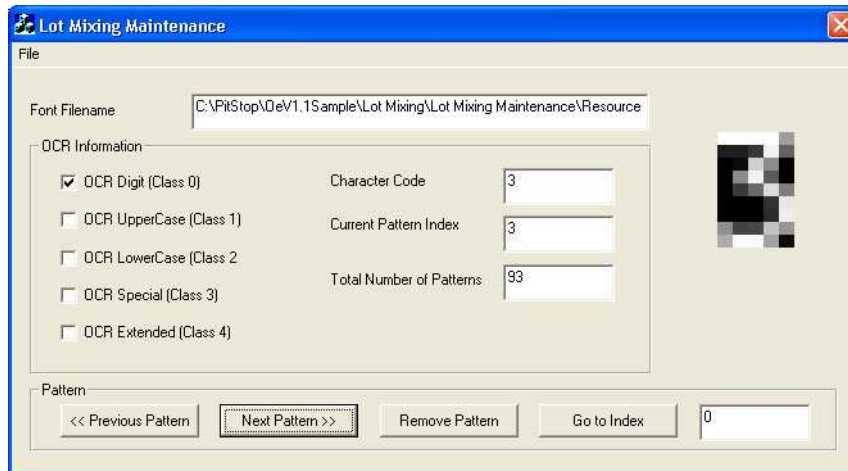
This application shows how to:

- Load an EasyOCR font file.
- Navigate through the stored patterns.
- Retrieve the associated character code and class.
- Remove a given pattern.



TIP

Each pattern in the font file is identified by its index.



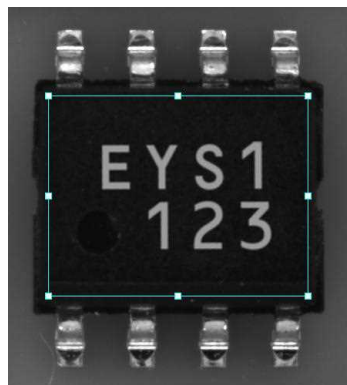
To use this example application:

1. Open the **File** menu.
2. Click on **Load Font File** to load a font file.
3. Click on **Save Font File** to save the font file (use the filename OCR-B.ocr in this example).

Checking lot mixing online

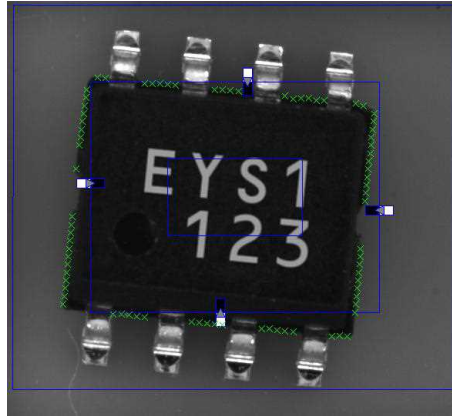
Initializing the application

1. The application loads the required font file for characters recognition (OCR-B.ocr in this example).
2. It performs the OCR operation only on the device body area (by means of an ROI).

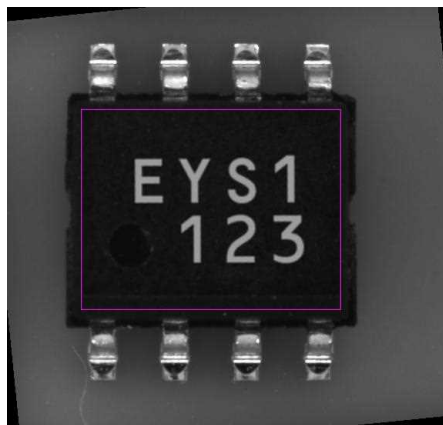


Inspecting the devices

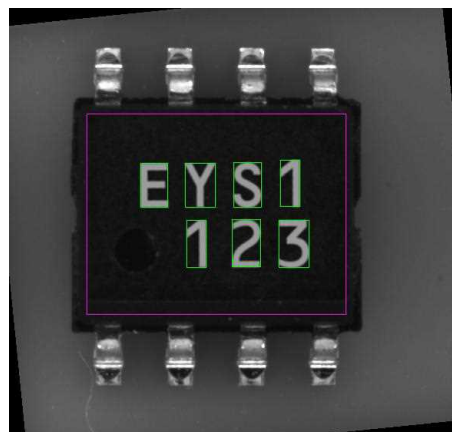
1. As the position of the device varies from one image to another, the application uses an `ERectangleGauge` tool to retrieve the position and the rotation angle of the device body.



2. The application registers and rotates the image to center the device body in the image. This is done on a temporary image storage.
3. The application positions an ROI on the device body area.



4. The application calls the EasyOCR recognition function within the defined ROI area.



5. The application displays the recognized string together with the original source image.

