

Sketch-to-Spice converter: Installation guide

Tarek Tohme

May 7th, 2018

1 Step 1: Downloading and installing OpenCV

Sketch-to-Spice requires OpenCV to run. Here are links to tutorials on how to install it for linux and Mac.

Linux Follow this [link](#)

Mac If you have homebrew, you can run the following command:

brew install opencv

from the terminal.

if not, you can install it by following this [tutorial](#).

2 Step 2: Using the binary

After unzipping the Sketch-to-Spice folder, you can run the program from the terminal window using ./symboldetector.

3 Step 3: Building the binary from source code (optional)

An easy way to build the binary from the cpp files is to use CMake. If you don't have it installed, you can get it from the [CMake download page](#).

After setting up CMake and OpenCV, create a text file called CMakeLists.txt, with the following contents (replacing the name of the executable and with symboldetector)

```
1 cmake_minimum_required(VERSION 2.8)
2 project( DisplayImage )
3 find_package( OpenCV REQUIRED )
4 include_directories( ${OpenCV_INCLUDE_DIRS} )
5 add_executable( DisplayImage DisplayImage.cpp )
6 target_link_libraries( DisplayImage ${OpenCV_LIBS} )
```

To generate the executable, just run the following commands:

```
1 cd <DisplayImage_directory>
2 cmake .
3 make
```

4 Training your own classifier

[This link](#) shows you how to train your own cascade classifier. I wrote a tool that simplifies the process of generating training images. You can find it in the classifier folder, under the name extractimages. The binary can be built and run by following the same steps as for the symboldetector.

Here's how to use it:

- Draw about 150 symbols of the component you want to train the classifier for on a piece of A4 paper.
- Scan the paper
- run extractimages with the path of the scanned image as the first argument on the command line, and the name of the component as the second argument. For example:

```
classifier ~> ./extractimages component_source_images/inductor.jpeg inductor
```

You should see a folder containing cropped binary images of each component drawn on the paper. The tool also produces the info file and a file containing the paths to the images, which are used in the tutorial to train the classifier. You can also use the tool to produce the negatives, or background images mentioned in the tutorial.

After training a classifier, you can input the path of the cascade.xml file to the application to enable detection for more components. Read the Help paragraph in the application for more info. You can use the following recommended settings to train the classifier:

```
-numPos 140  
-numNeg 600  
-numStages 5  
-w 100  
-h 25  
-maxFalseAlarmRate 0.1
```

5 Try it out!

We suggest you try the application out with the test5.jpeg image, located in the test_set folder. The object detector is quite sensitive to lighting, and it was trained using the author's own style of drawing, so getting it to detect other styles of components might be tricky. Also, it can only detect resistors, as this is a free trial version.