Computer Vision

Lab 1

Gavan Burke

Overview:

Develop a program to check bottles of glue and determine if there is a label on each bottle.

Plan of action:

In a recent tutorial, Dr. Dawson mentioned that getting the Hue, Luminance and Saturation of the image and then getting the Standard Deviation of that image would be the best approach to a problem like this.

How this was achieved:

I used Dr. Dawson's previous code for loading the images. By separating the glue bottles into an image per single glue bottle, this would make life a little bit easier while trying to see if it had a label or not.

Cropping the images further:

This part was possibly the most time consuming part of my program. As it was, I would not be able to see if the bottle had a label on it or not. This is because of the black background. The background was throwing the values of the standard deviation way off, so I could not get a proper read on them.

My goal was to crop each image so that I would only be looking at the label, or where the label should be, and get the standard deviation of that.

As each glue bottle was in a different position in each image, I had to hardcode each position in. This resulted in a big chunk of code which unfortunately looks a bit ugly.

The end result however was each bottle was cropped with only the body of the glue bottle on show, with as much of the black background cut off as I could.

Expected Results:

As we as humans can visually see which bottle has a label and which doesn't, a boolean array of the expected results was created. True meant a label was present, false otherwise.

Cut off:

To get the cut off, I took two non-labelled glue bottles that I thought would be the ones with the highest standard deviation based on how dark they looked in their original images.

A comparison of the standard deviation of both images returned the higher of the two. This was the value that was used for the cut off.

Converting to HLS and recording results:

The function cvColor along with with the code CV_BGR2HLS converts a color image into a Hue, Luminance and Saturation image. To get the right results, one would need to split this image into the three channels. This was done by using the built in function ‘split’.The channels are as follows:

hls_place[0] - Hue.

hls_plane[1] - Luminance.

hls_plane[2] - Saturation.

Experiments with the hue image were unsuccessful. Luminance was the next one to try and this provided better results.

If the image was less than the cut off, that means that there is no label on the bottle and the value in the array actualResults remains as false. Else, the index value would now be true.

The program then checks the value in the index of the actualResults array, and prints off whether or not there is a label present or not accordingly.

Compare Results (Metrics and Performance):

This function has two parameters, the expected results and the actual results from running this program.

The two are compared to see if the proper results have been achieved.

From analysing Dr. Dawson’s book, *Practical Introduction to Computer Vision With OpenCV, Section 8.6.3*, I found that I need to find the values for the True Positives, False Positives, True Negatives and False Negatives.

A quick comparison gave me these values and allowed me to perform the Metrics for the program.

Images:

Figure 1. Original Image

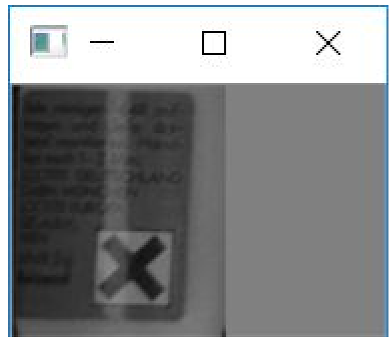
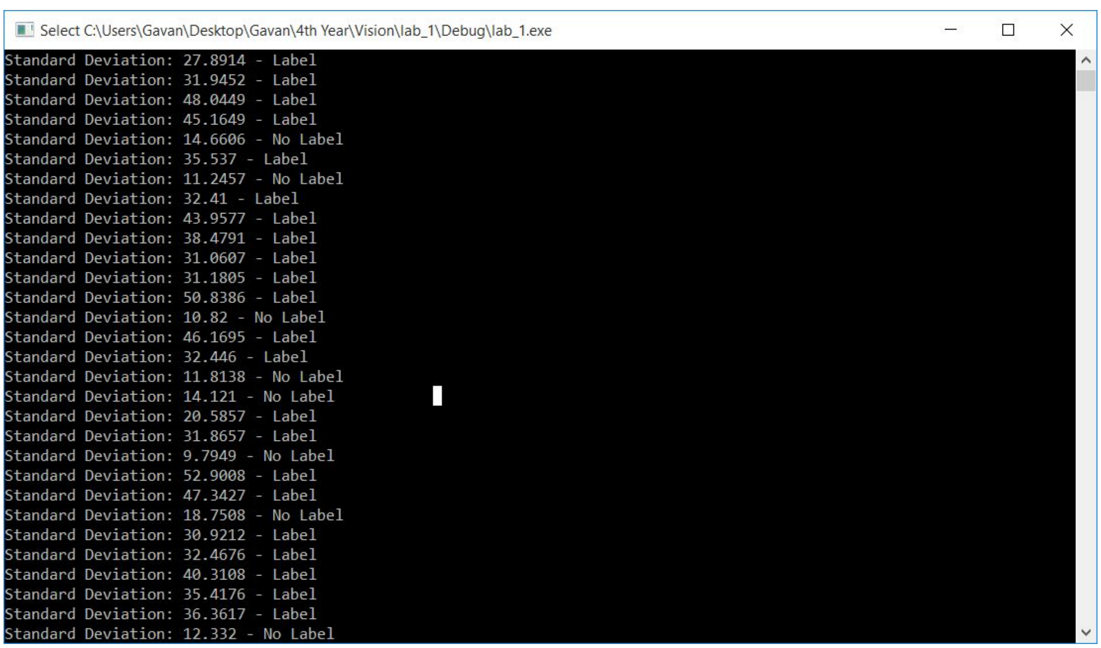


Figure 2. Cropped Luminance Image.



```
Select C:\Users\Gavan\Desktop\Gavan\4th Year\Vision\lab_1\Debug\lab_1.exe
Standard Deviation: 27.8914 - Label
Standard Deviation: 31.9452 - Label
Standard Deviation: 48.0449 - Label
Standard Deviation: 45.1649 - Label
Standard Deviation: 14.6606 - No Label
Standard Deviation: 35.537 - Label
Standard Deviation: 11.2457 - No Label
Standard Deviation: 32.41 - Label
Standard Deviation: 43.9577 - Label
Standard Deviation: 38.4791 - Label
Standard Deviation: 31.0607 - Label
Standard Deviation: 31.1805 - Label
Standard Deviation: 50.8386 - Label
Standard Deviation: 10.82 - No Label
Standard Deviation: 46.1695 - Label
Standard Deviation: 32.446 - Label
Standard Deviation: 11.8138 - No Label
Standard Deviation: 14.121 - No Label
Standard Deviation: 20.5857 - Label
Standard Deviation: 31.8657 - Label
Standard Deviation: 9.7949 - No Label
Standard Deviation: 52.9008 - Label
Standard Deviation: 47.3427 - Label
Standard Deviation: 18.7508 - No Label
Standard Deviation: 30.9212 - Label
Standard Deviation: 32.4676 - Label
Standard Deviation: 40.3108 - Label
Standard Deviation: 35.4176 - Label
Standard Deviation: 36.3617 - Label
Standard Deviation: 12.332 - No Label
```

Figure 3. Results from running the program.

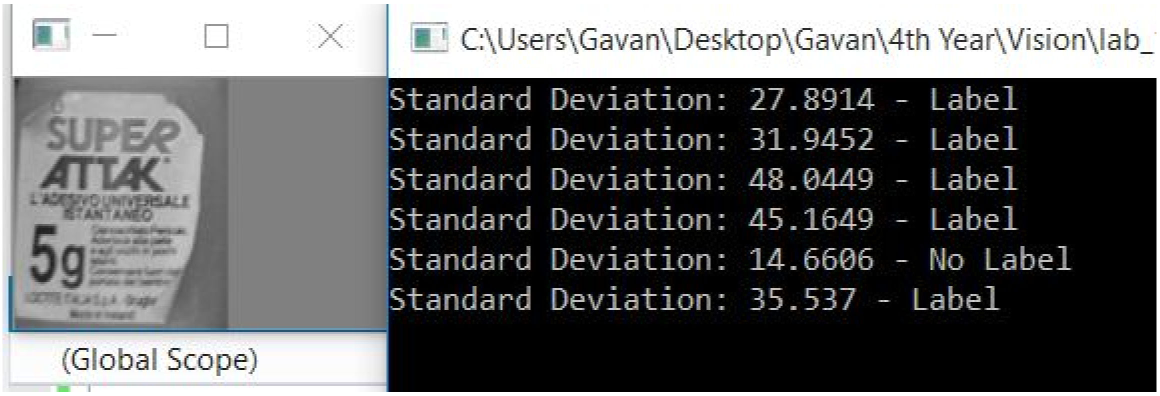


Figure 4. Current Image and results side by side.

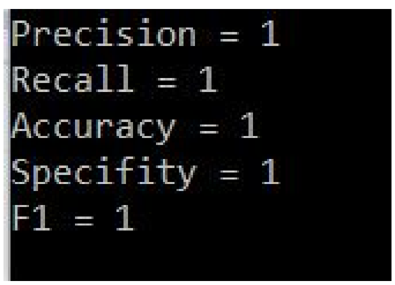


Figure 5. Metrics of program.

Formulae used for Metrics:

$$Recall = \frac{TP}{TP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Accuracy = \frac{TP + TN}{TotalSamples}$$

$$Specfcity = \frac{TN}{FP + TN}$$

$$F_\beta = (1 + \beta^2) \cdot \frac{Precision \cdot Recall}{(\beta^2 \cdot Precision) + Recall}$$