Final Project Report
ECE 231L; F. Viramontes

Bradley Evans
15 May 2020

ECE 231L Final Report on Project.

Bradley Evans for Francisco Viramontes, Our Instructor for ECE 231L.

I must say that I have learned quite a bit with my auxiliary project using Python code. The project was suggested to me by Prof Viramontes, because he knew I would be using Python code during this summer's internship at the AFRL Labs here at Kirtland AFB/Albuquerque, so I do believe that he wanted to prepare me for that experience. I think what I have learned by doing this project has been fruitful.

I had a Computer Vision class with Prof. Marios Pattichis here at UNM and received a good grade for it. But, for what I can remember of it, my gosh, I need a refresher course. That's what this [project has instilled in me: learning things that I should have with Dr. Pattichis almost a year ago. Let's begin with some rudimentaries of this project.

The first part of this project, a step by step effort, was to load and bring up an image for viewing, and to place that image in its original color format and in a gray-scale format. The Pythion program to do this is called "Show3Images.py", the results of which are shown in the folder "1.1 – Python Assignment 1 Folder".

Having accomplished this, the next task to accomplish was to place a solid circle around some aspect of the three, color images. First, a small, solid, red circle was placed around the right eye of the sunglasses of Tom Cruise. A larger, solid, blue circle was placed around the face of a Welsh rugby player, and finally, a larger, solid, green circle was placed around a colorful daffodil. The three Python programs to do this were "1_1_Solid_Circle_Cruise.py", "1_2_Solid_Circle_Daffodil.py", and "1_3_Solid_Circle_Rygbi_Player.py", respectively. These results are contained in the folder "2.1 – Solid Circle Folder."

The next stage was to find some other type of geometric object to place on an image. A clear circle image was chosen. A clear, red circle was placed around the left eye on the sunglasses of Tom Cruise; a clear, blue circle was placed around the face of a Welsh rugby player; and finally, a clear, green circle was placed around the image of a daffodil. The results for these are placed in the folder "2.2 – Clear Circle (Outline) Folder".

Lastly, for this geometric exercise, a rectangular, purple, bounding box was placed on the face of Tom Cruise (a.k.a. "Smiley"); a brown, rectangular bounding box was placed around the face of a Welsh rugby player; and finally, a green, rectangular, bounding box was placed around the perimeter of a daffodil. The results of this effort may be found in the folder "2.3 – Bounding Box Rectangle Folder".

Moving forward with our geometric shapes, all three of the shapes were placed on a single image, not three separate ones, as the above three paragraphs describe. Thus, a solid green circle, a clear, red circle, and a purple, rectangular, bounding box was placed on the face of Tom Cruise. Next, a solid blue circle, a clear, green circle, and a brown, rectangular, bounding box were placed on the faces of three different Welsh rugby players. Finally, a solid, grey circle, a green, clear circle, and a dark green, rectangular bounding box were placed on three different daffodils. Three different codes were written for the three different images obtained. The results of which were placed in the folder “4.1 – Three Shapes on One Image with Code Folder”.

Prof. Viramontes gave me an alternate way to display an image. This is accomplished by the following code:

```
# Display an Image.

title = 'Daffodils Covered by a Solid Circle, Clear Circle, and a Box'

print(title)

print(title+'.png')

cv2.imshow(title,img)
```

In other words, an assignment may be made for the title of an image. The image may be saved in the same “tricky” way:

```
# Use this function to save an image.

cv2.imwrite(title+'.png',img)
```

All in all, this is just another convenient way to display some repetitive code (a title) only a few lines down from its origination. The code and this new and “tricky” way of displaying an image is to be found in the folder “5.1 – Images Using Tricky Code Folder”.

Finally, for the “Grand-Daddy of them all” conclusion of this project, a Haar detection was supposed to be made of a face with some eyes. We had wrestled with all the key components of face detection and geometric shapes in the previous tasks as described above. We now had to implement some code which would bound and label the face and eye features of Mr. Cruise, the Welsh rugby players, and a daffodil. It is a sorrowful thing to report that this effort failed due to the reasons of the Python program’s inability to incorporate two XML files into its code. Both of those critical files are called “haarcascade_frontface_default.xml” and “haarcascade_eye.xml”. It is with these files, and these two files alone, which prevented the execution of our program. It was tough luck; we just need some more time to figure out exactly what went wrong and why these two XML files will not load into the main code. Drat! The results of theses efforts are found in the final folder “6.1 – Haar Detection for Face and Eyes Folder”, located in my GitHub repository.

In conclusion, this was a good exercise and a start on further understanding Computer Vision, which I will do this coming summer at the AFRL Labs here in Albuquerque. Instead of innocent photographs and images of Mr. Cruise, Welsh rugby players, and daffodils, other images such as thrusters, solar panels, retro rockets, and other items may be trained for our program to identify in space and other remote locations. This is only an initial step in that direction.

Thank you, Mr. Viramontes, for a good start in this direction.