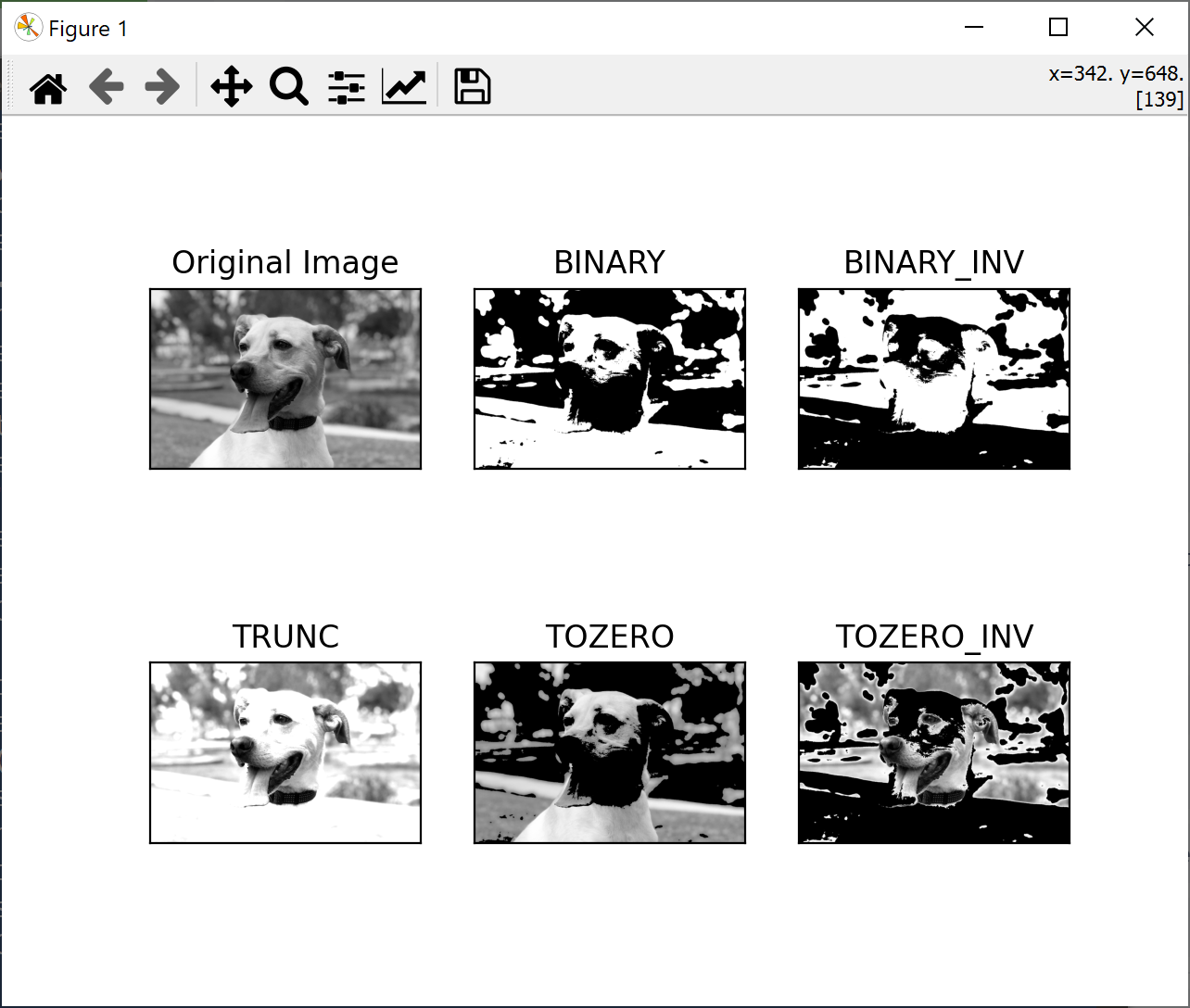
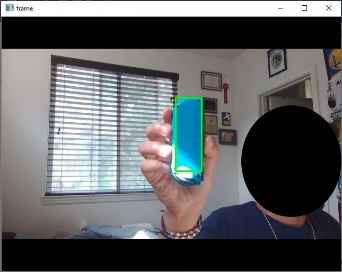
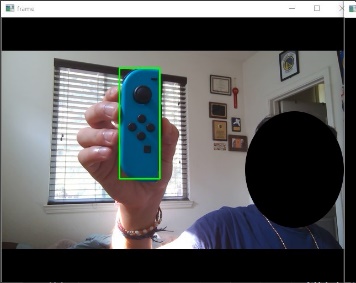
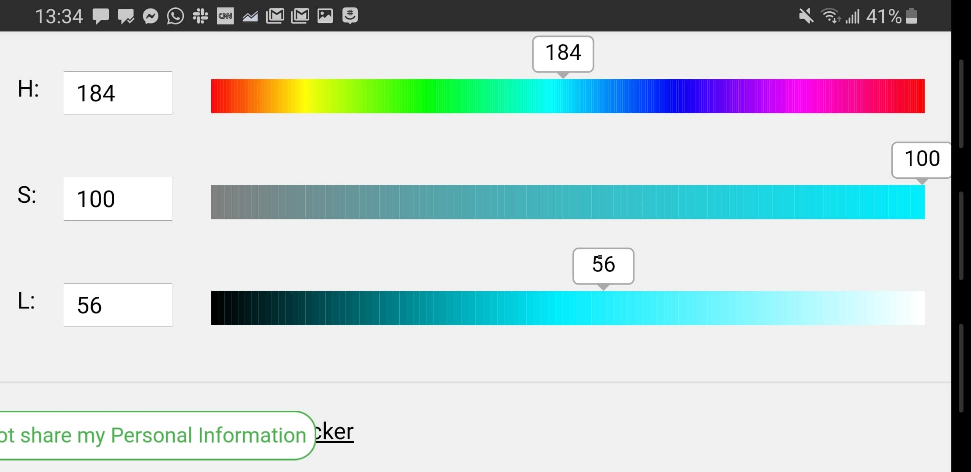
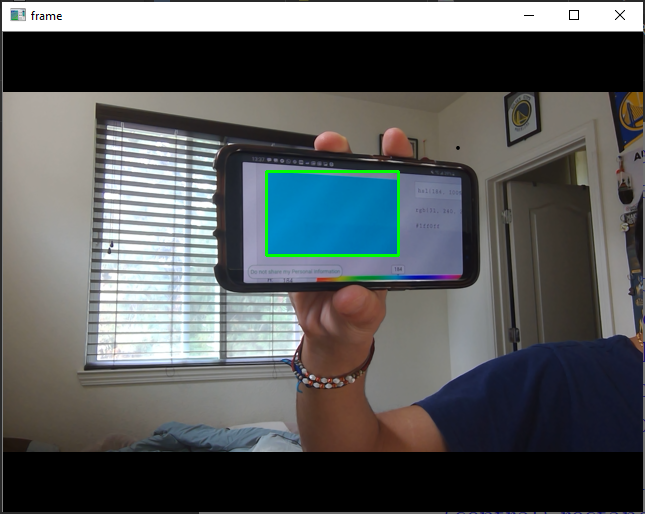
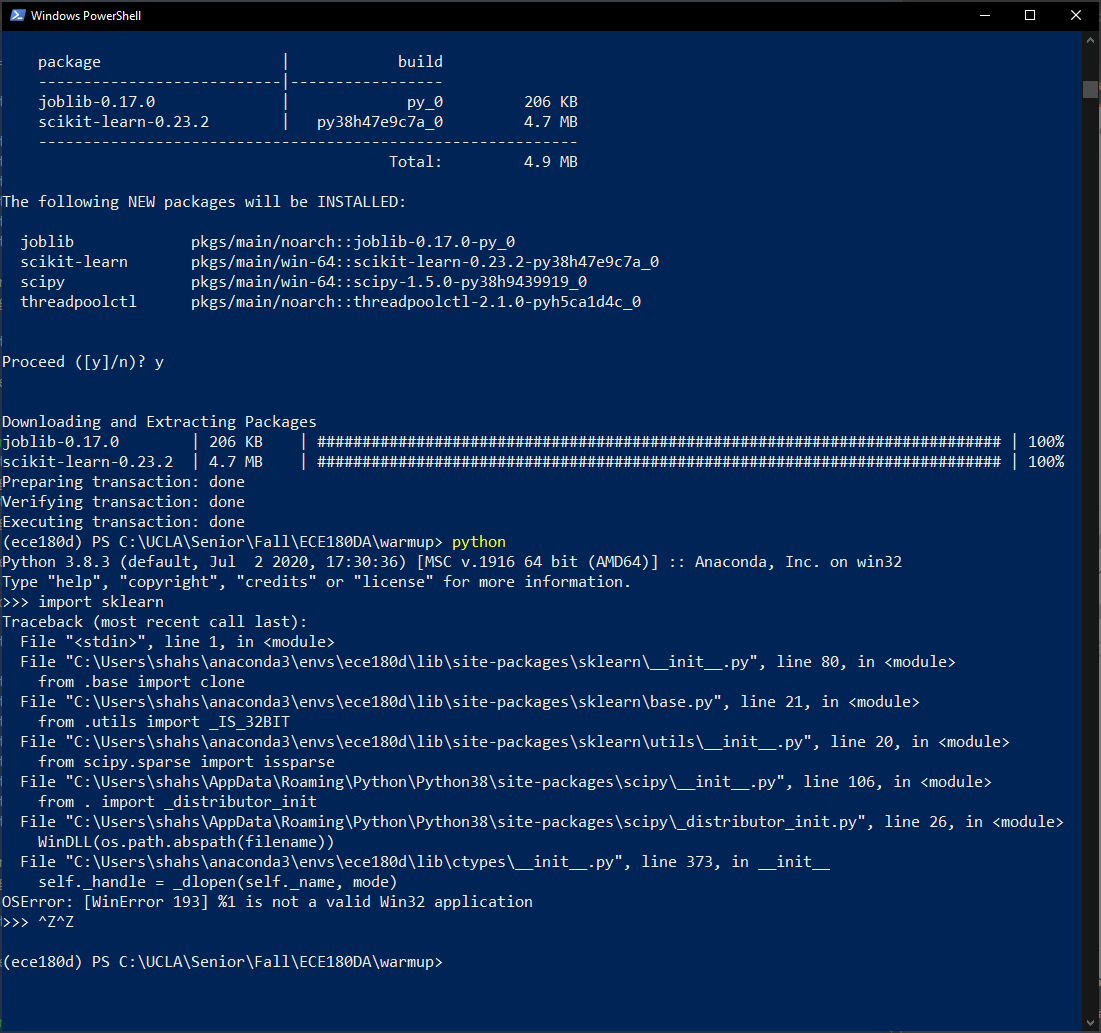
Shalin Shah

Professor Shoarinejad

8 October 2020

Week 0 Lab Report

1. What did you plan to do last week?
   1. Start and complete tutorial 1 – Introduction to SW Development and CV
   2. Familiarize with conda environments and true version control in Git
   3. Explore OpenCV libraries and complete one of Advanced Topics provided
2. What did you end up doing last week?
   1. Completed tutorial 1
      1. Have a conda instance named “ece180d” that I activate and deactivate
      2. Has all of the packages neatly downloaded there
      3. Did extensive research on version control in Git:
         1. [Atlassian](https://www.atlassian.com/git/tutorials/setting-up-a-repository)
         2. [YouTube Video](https://www.youtube.com/watch?v=SWYqp7iY_Tc)
      4. Followed OpenCV (cv2) tutorials given in sections 5
         1.  (one example of thresholding with dog picture from Google)
         2. One thing that our group can look into further is template matching and edge detection since we’re looking to compare two players’ silhouettes
   2. Answers to Task 4:
3. I was tracking a controller that’s relatively monochromatic and is a light blue and was able to do so (the first picture is with HSV and the second with BGR). As expected, HSV performed much better than RGB. In my room, there were subtle lighting changes at the time of doing this project and HSV was able to pick up all of them. BGR, on the other hand, struggled because I had to try to find the exact color value that associated with my controller, and often times that value doesn’t exist because of slight invariances in color. For that reason, I had a very broad range for individual BGR values: **lower = np.array([175, 140, 0]), upper = np.array([230,175,80])**, whereas for HSV: **lower = np.array([80, 100, 85]), upper = np.array([100,255,255])**, we see that while there’s a broad range for saturation and “value” (intensity), there’s a much smaller range for the hues that are needed to discern whether the object is there. For this reason, HSV often performs better in practice, especially in CV.
4. Yes, there is a major difference. If I shine a bright phone light directly at the blue controller that I used in the past example, I still am able to get the rectangle around the controller in the case of HSV, but not in the case of BGR. This is due to the fact that in BGR, the controller looks “white” because BGR doesn’t recognize that changes in lighting don’t change the underlying image. HSV does recognize this, and since we have saturation and light intensity accounted for, we can tell that the same colors (in the hue range of [80,100]) are present even with major changes in lighting. There’s a slight difference in tracking, mostly because BGR often isn’t continuous in its recognition of the controller
5. The first screenshot is from my phone showing the HSL values that I chose:  
     
     
     
     
     
     
     
     
     
     
   The second screenshot is from my phone showing the zoomed in version of the color output:  
     
     
     
     
     
     
   The next screenshot is how my existing HSV implementation was able to pick it up:  
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
   The BGR implementation was not able to pick it up with the consistency that this one was.
6. For this part, I wasn’t able to get the package dependencies to work out in my favor. I wasn’t able to get scikit-learn to download properly in my virtual environment. This will require me to dig in a bit more and try alternate methods including another OS (Linux instead of my existing Windows PowerShell conda environment)   
     
   This is the error I get: 
7. This next week:
   1. Fix issues from this week (with virtual environment not having scikit-learn)
   2. Finish Task 4.4.
      1. I’ve already followed the sample code given
      2. Now just need to implement the rectangle portion of code
   3. Discuss plans for project further with team
   4. Complete tutorial for this week (introduction to HW)