Bhuvan Jammalamadaka

Professor Nitin Sanket

FIRE198 AUS

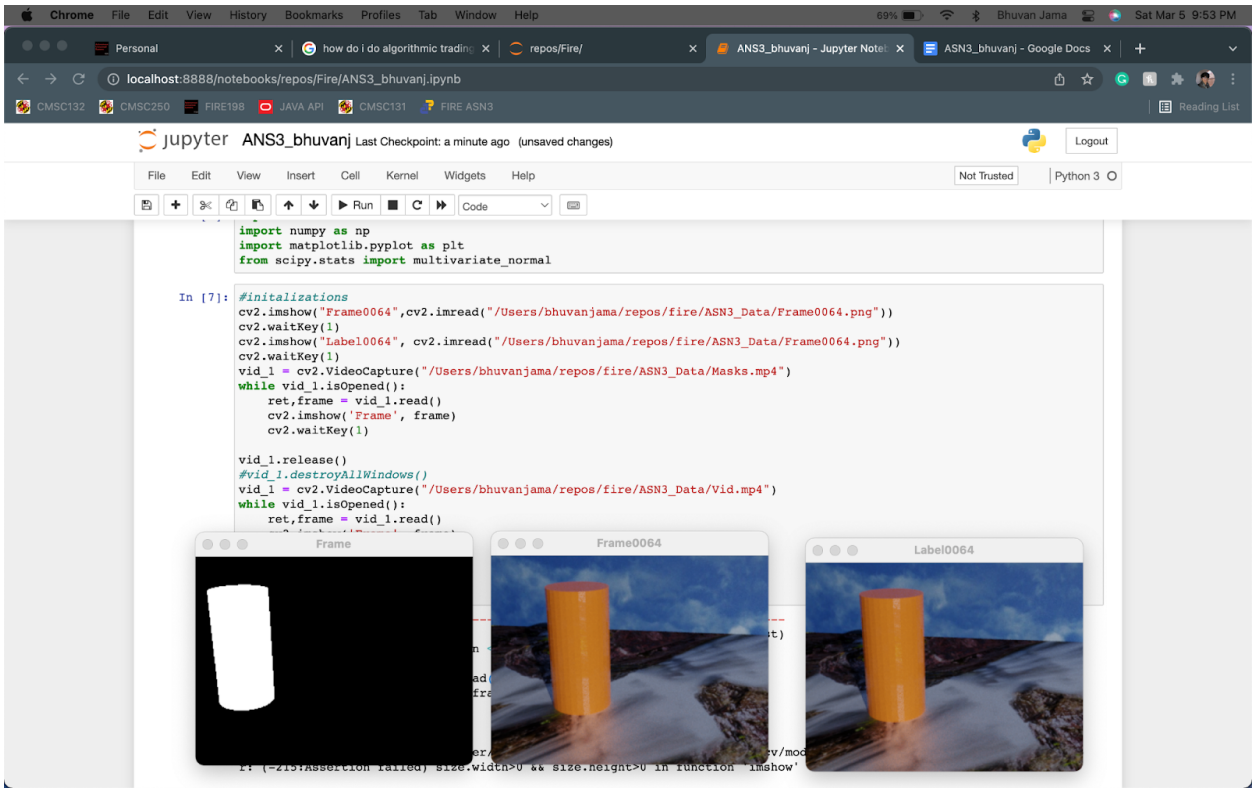
4 March 2022

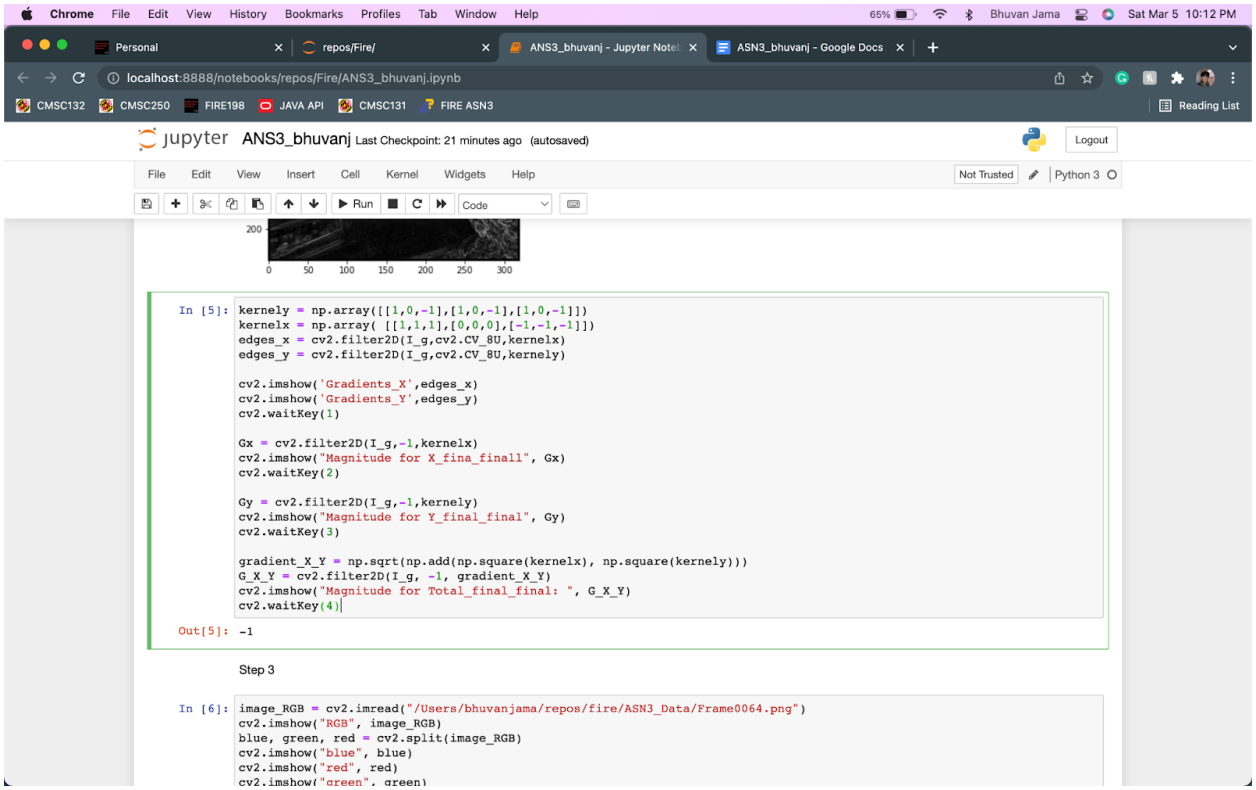
ASN3: How does a Robot See the World?

Step 2:

cv2.imread(file path) → to read an image

cv2.imshow() → to show the image



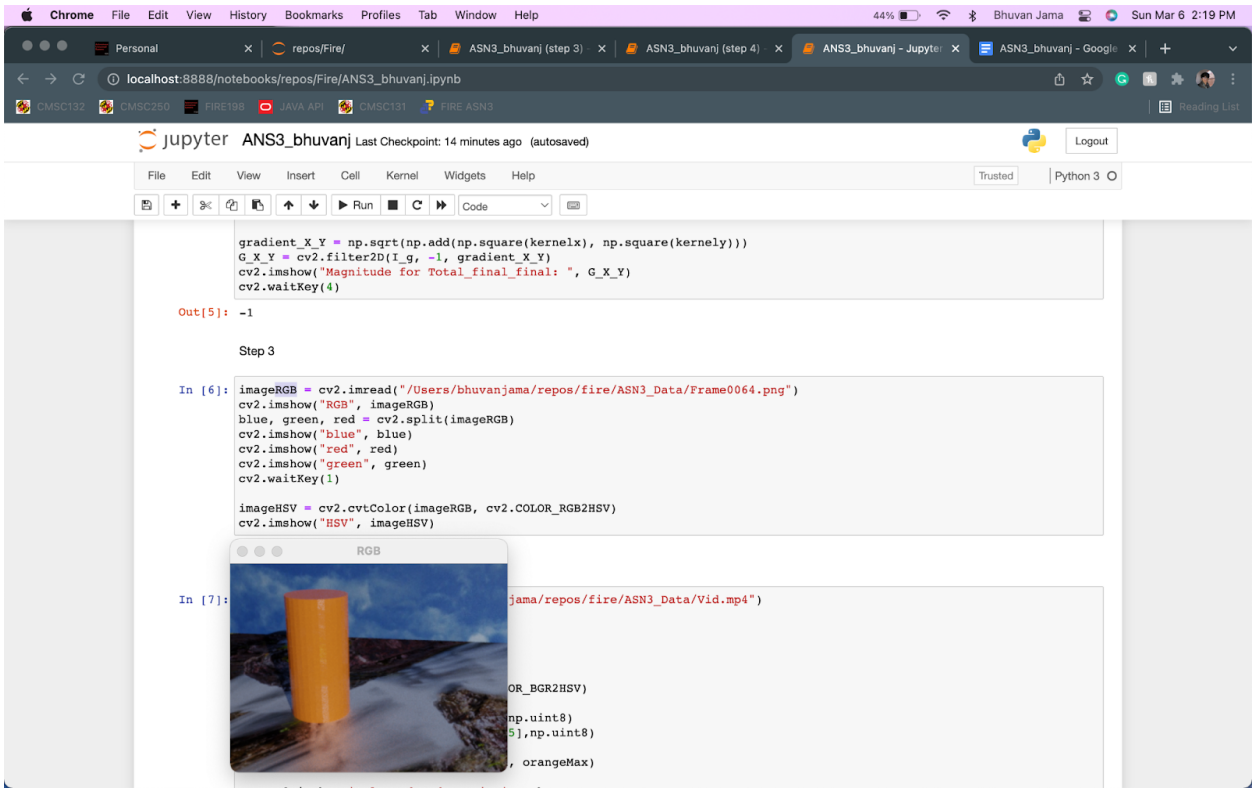


```
In [5]: kernely = np.array([[1,0,-1],[1,0,-1],[1,0,-1]])
kernelx = np.array( [[1,1,1],[0,0,0],[-1,-1,-1]])
edges_x = cv2.filter2D(I_g,cv2.CV_8U,kernelx)
edges_y = cv2.filter2D(I_g,cv2.CV_8U,kernely)

cv2.imshow('Gradients_X',edges_x)
cv2.imshow('Gradients_Y',edges_y)
cv2.waitKey(1)

Gx = cv2.filter2D(I_g,-1,kernelx)
cv2.imshow("Magnitude for X_fina_finall", Gx)
cv2.waitKey(2)

Gy = cv2.filter2D(I_g,-1,kernely)
cv2.imshow("Magnitude for Y_final_final", Gy)
cv2.waitKey(3)

gradient_X_Y = np.sqrt(np.add(np.square(kernelx), np.square(kernely)))
G_X_Y = cv2.filter2D(I_g, -1, gradient_X_Y)
cv2.imshow("Magnitude for Total_final_final: ", G_X_Y)
cv2.waitKey(4)

Out[5]: -1
```

Step 3

```
In [6]: image_RGB = cv2.imread("/Users/bhuvanjama/repos/fire/ASN3_Data/Frame0064.png")
cv2.imshow("RGB", image_RGB)
blue, green, red = cv2.split(image_RGB)
cv2.imshow("blue", blue)
cv2.imshow("red", red)
cv2.imshow("green", green)
```

Step 3:



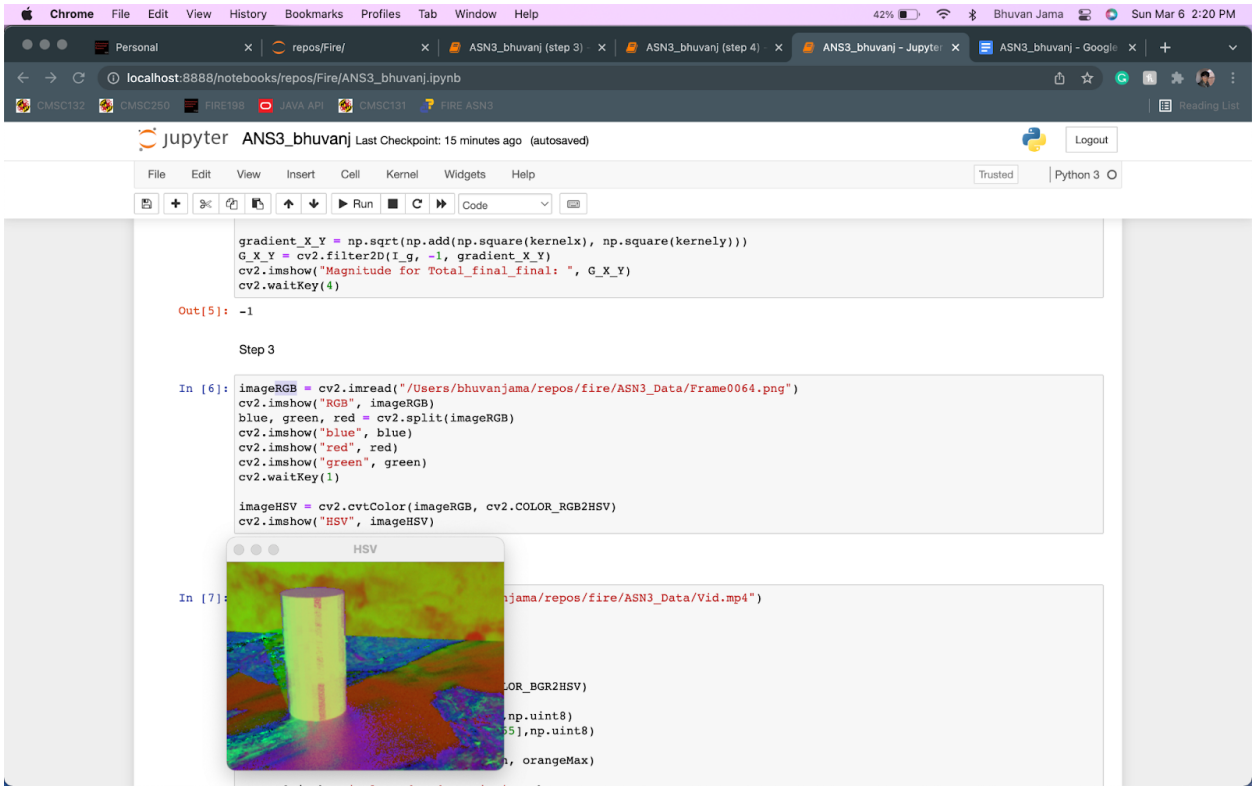
```
gradient_X_Y = np.sqrt(np.add(np.square(kernelx), np.square(kernely)))
G_X_Y = cv2.filter2D(I_g, -1, gradient_X_Y)
cv2.imshow("Magnitude for Total_final_final: ", G_X_Y)
cv2.waitKey(4)

Out[5]: -1
```

Step 3

```
In [6]: imageRGB = cv2.imread("/Users/bhuvanjama/repos/fire/ASN3_Data/Frame0064.png")
cv2.imshow("RGB", imageRGB)
blue, green, red = cv2.split(imageRGB)
cv2.imshow("blue", blue)
cv2.imshow("red", red)
cv2.imshow("green", green)
cv2.waitKey(1)

imageHSV = cv2.cvtColor(imageRGB, cv2.COLOR_RGB2HSV)
cv2.imshow("HSV", imageHSV)
```

RGB

```
In [7]: jama/repos/fire/ASN3_Data/Vid.mp4")

OR_BGR2HSV)

np.uint8)
5],np.uint8)

, orangeMax)
```



What differences do you visually observe between the RGB and HSV images? Why is one likely better than the other?

Visually the RGB version highlights the orange more "clearly" than the HSV because there is a more distinct difference in the pixels of the image. For this reason, I think the RGB version would be better because it helps the model more easily classify the orange that it is looking for. An interesting observation I made was the way light is pointed out in both images. In the HSV you can more clearly see that the middle of the cylinder is a different shade because of the lighting factor, however, on the RGB image, it is not highlighted as distinct of a difference.

Step 4:

What factors make the same object color appear different on the captured image? List five factors (feel free to list more).

There are several factors that will influence the color appearance of a captured image, these include the shape of the object, hue variation, reflectance of the shape, material of the object, and the environment.

What are the issues with the simple color threshold you implemented in the previous step?

The issues with the simple color threshold implemented in the previous step are that the output results are not as clear because the background of the image is not a constant color that can be classified as one value (making it easier for the "orange" to be identified). Rather, there are several color gradients embedded in the picture making the orange cone not as obviously identified as a more effective method. In other words, the factors that influence an object (the ones stated above) affect the way the image is classified when using a simple color threshold.

Is this method any better than the first method?

The second method is better than the first method because there is a higher chance of classifying the image's color by attributing a probability aspect to the value of the pixel rather than giving it a binary value of "orange" or "not orange". The gaussian allows for the sum of these probabilities to be averaged out and leads to a more accurate value even though the other variables are more skewed.

What are the issues in the second method? How can you fix them?

The second method does not take into account the color of the environment making it inaccurate when the lighting is varied. You can fix this by using a "fancy function" which can be computed by the summation of all the Gaussians produced. By taking the log of the fancy function machine learning engineers are able to find whether the function is entirely nonincreasing or nondecreasing.

What color space did you choose and why?

I used the HSL colorspace because it expresses orange the most effectively. This allows for the model to be trained to pick up the color better as it has a more clear idea of what different values of “orange” are.

Challenges you faced in solving this assignment along with lessons learned.

I think the largest challenge with this assignment was the learning curve associated with this subject. I am strongly interested in it, however, balancing this with the workload from my other technical classes (3) was a tad challenging. Time management is a factor but even after effective strategies of compartmentalizing my time I had to use two late passes on this assignment. I believe that is okay because I learned a lot from this assignment and I would rather turn it in late than turn it in blank. Plus, the late passes came in very clutch for me as we have 6 of them throughout the semester.

Positive or Negative feedback about this assignment if you have any.

I feel that it was a fair assignment because this field has a very steep learning curve and it is important to understand what we are learning/doing (as you had mentioned in class). The late passes coupled with the extension allowed for a lot of leeways and definitely helped me a lot. I also feel that the office hours, PRM’s, and fellow FIRE students were all great resources to seek help and learn more.