Guided Assignment On Unsupervised Learning

Name: **Shreya Shrivastava**

Course: **Artificial Intelligence And Machine Learning**

Batch Four

Duration: 12 Months

Problem statement:Using OpenCV implement a single object tracker. Steps to be implemented:

- **a) Use a pre-recorded video or your webcam to have a video Capture object.**
- **b) Mark the region of interest (ROI or the object you want to track) using it coordinates in the first frame.**
- **c) Calculate the histogram of the ROI.**
- **d) Iteratively calculate the histogram at each location (using cv2,calcBackProject)
and then apply mean shift to get the updated location of the ROI.**

Prerequisites:

The libraries as well as things required in order for the program to work:

I. **Python 3.6** : The following url https://www.python.org/downloads/ can be referred to download python. Once you have python downloaded and installed, you will need to setup PATH variables (if you want to run python program directly, detail instructions are below in how to run software section). To do that check this: https://www.pythoncentral.io/add-python-to-path-python-is-not-recognized-as-an-internal-or-external-command/. Setting up PATH variable is optional as you can also run program without it and more instruction are given below on this topic. Second option is to download anaconda and use its anaconda prompt to run the commands. To install anaconda check this url : https://www.anaconda.com/download/

II. **OpenCV** **:** OpenCV can be downloaded from the following url: https://sourceforge.net/projects/opencvlibrary/. It is strongly recommended to download OpenCV in a virtual environment.

III. ADDITIONAL PACKAGES : You will also need to download and install below 3 packages- numpy, scikit learn and matplotlib after you install either python or anaconda from the steps above. If you have chosen to install python 3.6,then run the following commands in command prompt/terminal to install these packages :

MATPLOTLIB: pip install -U matplotlib

If using Anaconda then run the following commands in anaconda prompt to install these packages:

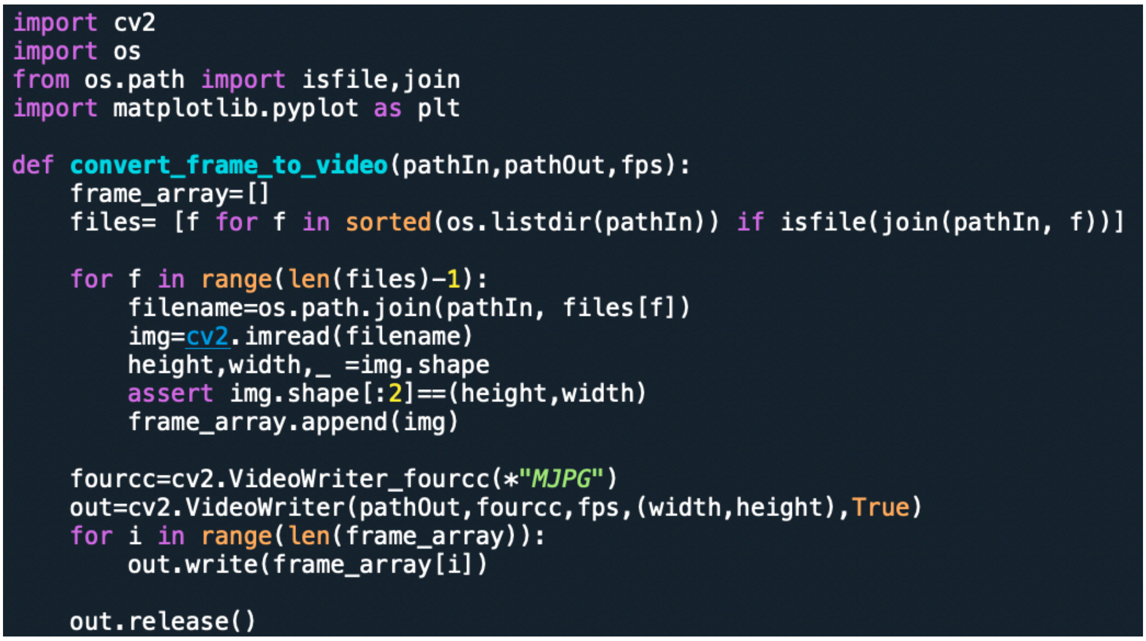
MATPLOTLIB: conda install -c anaconda matplotlib

IV. METHODS USED:

A. MEAN SHIFT ALGORITHM

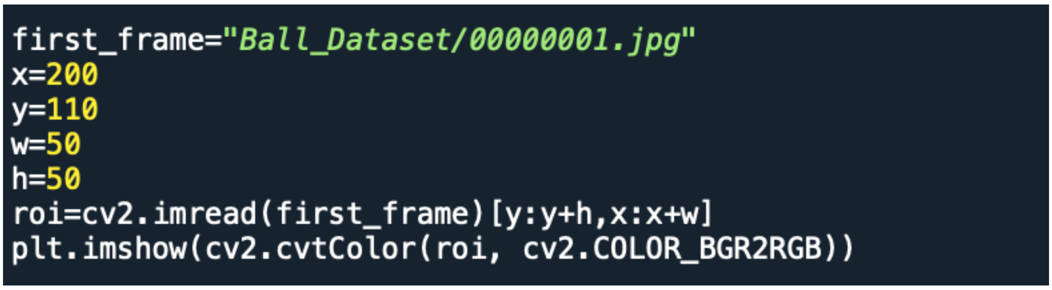
THE PROJECT :

1. Importing the libraries and writing a simple functions to take in the frames and convert them into a video using cv2.



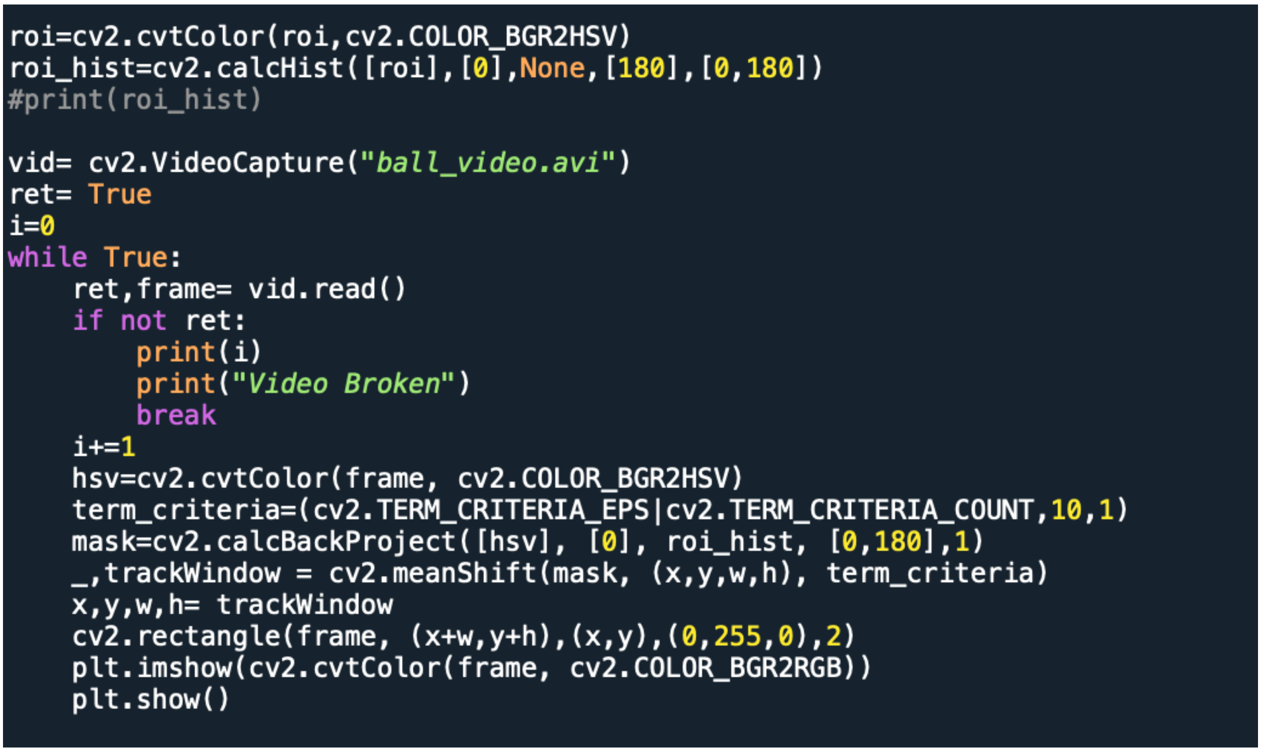
```
import cv2
import os
from os.path import isfile,join
import matplotlib.pyplot as plt

def convert_frame_to_video(pathIn,pathOut,fps):
    frame_array=[]
    files= [f for f in sorted(os.listdir(pathIn)) if isfile(join(pathIn, f))]

    for f in range(len(files)-1):
        filename=os.path.join(pathIn, files[f])
        img=cv2.imread(filename)
        height,width,_ =img.shape
        assert img.shape[:2]==(height,width)
        frame_array.append(img)

    fourcc=cv2.VideoWriter_fourcc(*"MJPG")
    out=cv2.VideoWriter(pathOut,fourcc,fps,(width,height),True)
    for i in range(len(frame_array)):
        out.write(frame_array[i])

    out.release()
```

2. Detecting the object in the initial frame and calculating our ROI.

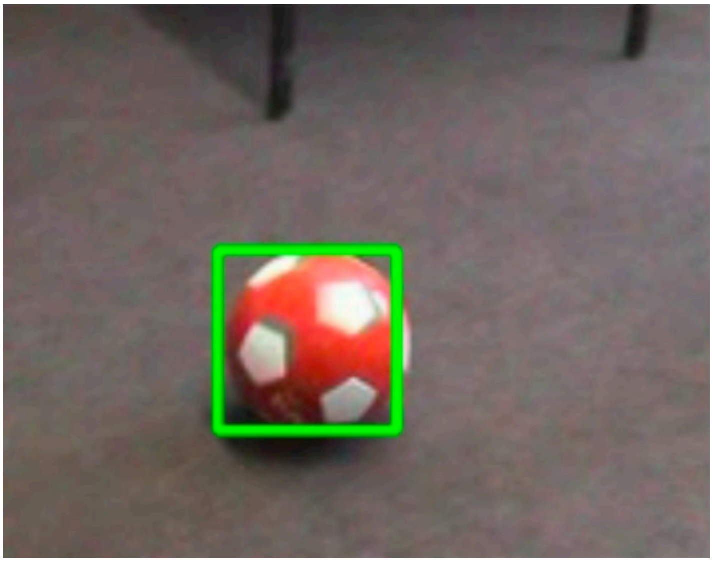


```
first_frame="Ball_Dataset/00000001.jpg"
x=200
y=110
w=50
h=50
roi=cv2.imread(first_frame)[y:y+h,x:x+w]
plt.imshow(cv2.cvtColor(roi, cv2.COLOR_BGR2RGB))
```

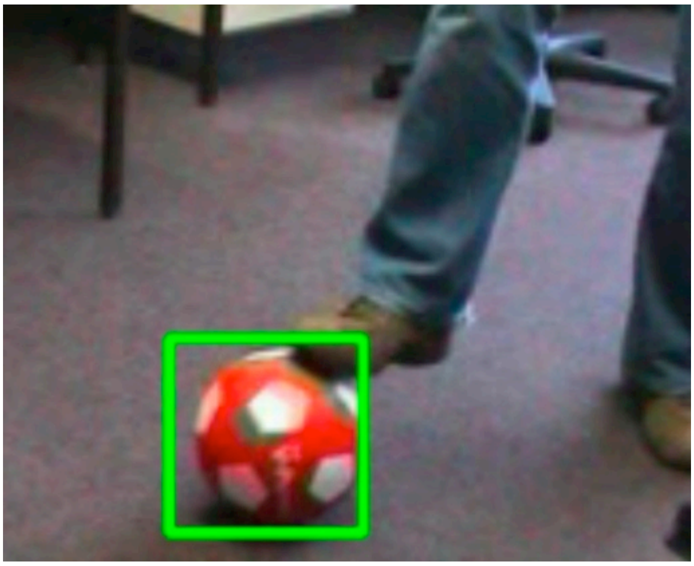
4. Calculating the histogram of our ROI at each location and applying mean shift to get the updated location of the object.



```
roi=cv2.cvtColor(roi,cv2.COLOR_BGR2HSV)
roi_hist=cv2.calcHist([roi],[0],None,[180],[0,180])
#print(roi_hist)

vid= cv2.VideoCapture("ball_video.avi")
ret= True
i=0
while True:
    ret,frame= vid.read()
    if not ret:
        print(i)
        print("Video Broken")
        break
    i+=1
    hsv=cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
    term_criteria=(cv2.TERM_CRITERIA_EPS|cv2.TERM_CRITERIA_COUNT,10,1)
    mask=cv2.calcBackProject([hsv], [0], roi_hist, [0,180],1)
    _,trackWindow = cv2.meanShift(mask, (x,y,w,h), term_criteria)
    x,y,w,h= trackWindow
    cv2.rectangle(frame, (x+w,y+h),(x,y),(0,255,0),2)
    plt.imshow(cv2.cvtColor(frame, cv2.COLOR_BGR2RGB))
    plt.show()
```



First Frame



FINAL RESULTS