Mean-Shift: Single Object Tracking in Images

Name - Amith Korada

Course - AI & ML
(Batch - 4)

Duration - 12 Months

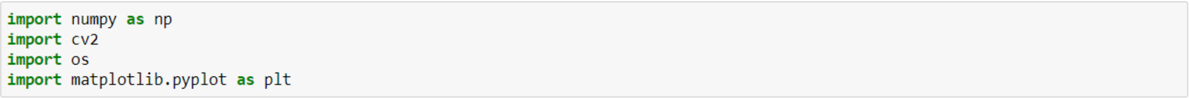
Problem Statement - Building a Machine Learning model to track a single object.
Prerequisites -

What things you need to install the software and how to install them:

Python 3.6 This setup requires that your machine has the latest version of python. The following URL https://www.python.org/downloads/ can be referred to as download python.

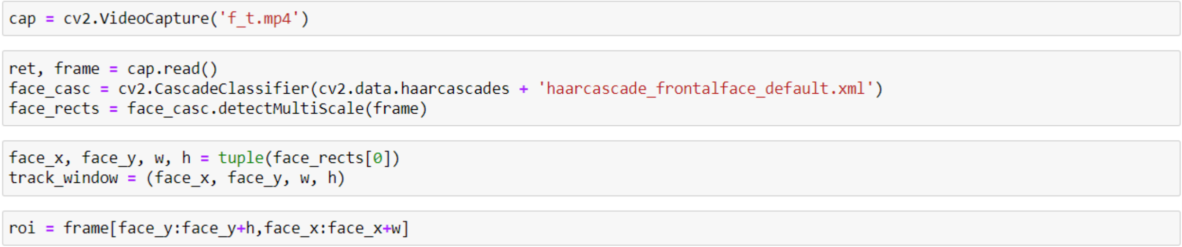
The second and easier option is to download anaconda and use its anaconda prompt to run the commands. To install anaconda check this URL https://www.anaconda.com/download/ You will also need to download and install the below 3 packages after you install either python or anaconda from the steps above Sklearn (scikit-learn) numpy scipy if you have chosen to install python 3.6 then run the below commands in command prompt/terminal to install these packages pip install -U sci-kit-learn pip install NumPy pip install scipy if you have chosen to install anaconda then run the below commands in anaconda prompt to install these packages conda install -c sci-kit-learn conda install -c anaconda numpy conda install -c anaconda scipy.

1. Importing necessary libraries-



```
import numpy as np
import cv2
import os
import matplotlib.pyplot as plt
```

2. Loading the recorded video and marking the ROI -



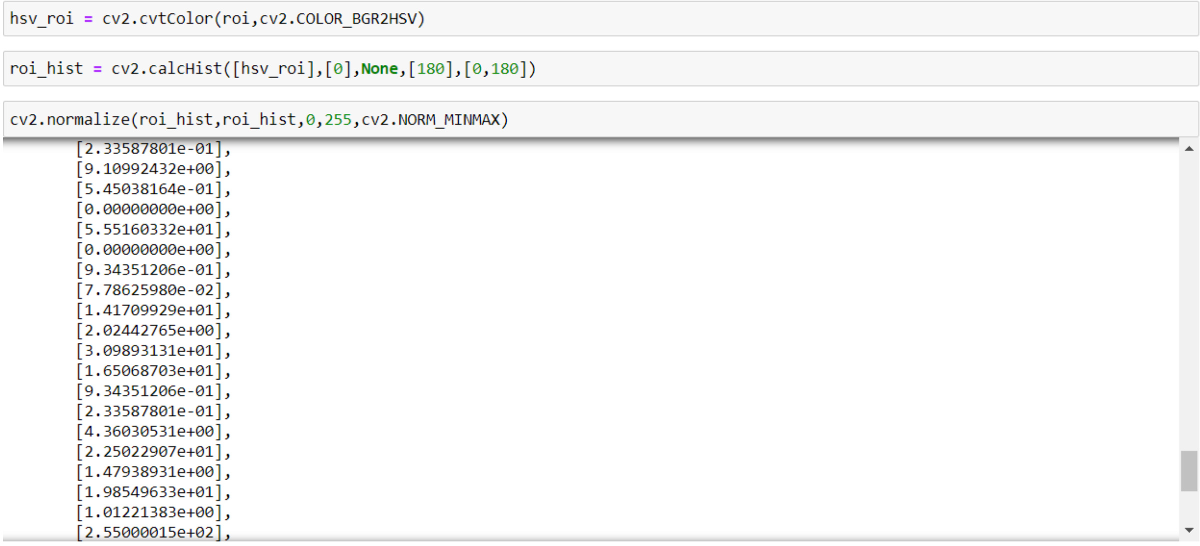
```
cap = cv2.VideoCapture('f_t.mp4')
```

```
ret, frame = cap.read()
face_casc = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')
face_rects = face_casc.detectMultiScale(frame)
```

```
face_x, face_y, w, h = tuple(face_rects[0])
track_window = (face_x, face_y, w, h)
```

```
roi = frame[face_y:face_y+h,face_x:face_x+w]
```

3. Calculating the histogram of ROI and Normalising it -



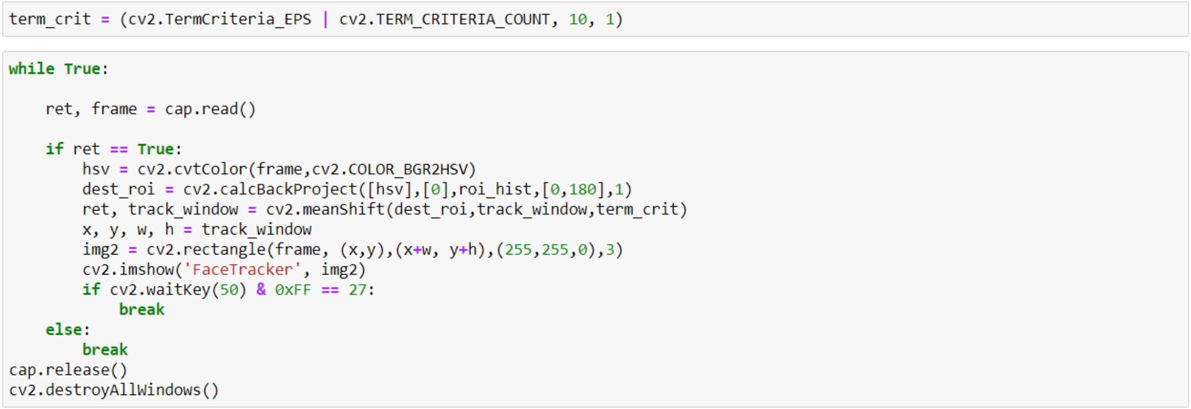
```
hsv_roi = cv2.cvtColor(roi,cv2.COLOR_BGR2HSV)
```

```
roi_hist = cv2.calcHist([hsv_roi],[0],None,[180],[0,180])
```

```
cv2.normalize(roi_hist,roi_hist,0,255,cv2.NORM_MINMAX)
```

```
       [2.33587801e-01],
       [9.10992432e+00],
       [5.45038164e-01],
       [0.00000000e+00],
       [5.55160332e+01],
       [0.00000000e+00],
       [9.34351206e-01],
       [7.78625980e-02],
       [1.41709929e+01],
       [2.02442765e+00],
       [3.09893131e+01],
       [1.65068703e+01],
       [9.34351206e-01],
       [2.33587801e-01],
       [4.36030531e+00],
       [2.25022907e+01],
       [1.47938931e+00],
       [1.98549633e+01],
       [1.01221383e+00],
       [2.55000015e+02],
```

4. Single Object Tracking



```
term_crit = (cv2.TermCriteria_EPS | cv2.TERM_CRITERIA_COUNT, 10, 1)
```

```
while True:

    ret, frame = cap.read()

    if ret == True:
        hsv = cv2.cvtColor(frame,cv2.COLOR_BGR2HSV)
        dest_roi = cv2.calcBackProject([hsv],[0],roi_hist,[0,180],1)
        ret, track_window = cv2.meanShift(dest_roi,track_window,term_crit)
        x, y, w, h = track_window
        img2 = cv2.rectangle(frame, (x,y),(x+w, y+h),(255,255,0),3)
        cv2.imshow('FaceTracker', img2)
        if cv2.waitKey(50) & 0xFF == 27:
            break
    else:
        break
cap.release()
cv2.destroyAllWindows()
```