FACE AND EYE DETECTION USING HAAR CASCADES

The key aspect in face recognition is detecting relevant features in human face like eyes, eyebrows, nose, lips. So how do we detect these features in real time/in an image ? The answer is **Haar Wavelets or Haar Features**. And the algorithm used is called as Viola-Jones Algorithm.Haar features are sequence of rescaled square shape functions proposed by Alfred Haar in 1909. They are similar to convolution kernels in the Convolution Neural Networks. We will apply these haar features to all relevant parts of face so as to detect human face.

It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images.

Here we will work with face detection. Initially, the algorithm needs a lot of positive images (images of faces) and negative images (images without faces) to train the classifier. Then we need to extract features from it. For this, haar features shown in below image are used. They are just like our convolutional kernel. Each feature is a single value obtained by subtracting sum of pixels under white rectangle from sum of pixels under black rectangle



These features are stored in a .xml file which we will be using for face and eye detection.

CODE:

import numpy as np

import cv2

face\_classifier = cv2.CascadeClassifier('/content/drive/My Drive/HAAR/Haarcascades/haarcascade\_frontalface\_default.xml')

eye\_classifier = cv2.CascadeClassifier('/content/drive/My Drive/HAAR/Haarcascades/haarcascade\_eye.xml')

img = cv2.imread('/content/drive/My Drive/HAAR/APJ1.jpg')

gray = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)

faces = face\_classifier.detectMultiScale(gray, 1.3, 6)

# When no faces detected, face\_classifier returns and empty tuple

if faces is ():

    print("No Face Found")

for (x,y,w,h) in faces:

    cv2.rectangle(img,(x,y),(x+w,y+h),(127,0,255),2)

    cv2\_imshow(img)

    cv2.waitKey(0)

    roi\_gray = gray[y:y+h, x:x+w]

    roi\_color = img[y:y+h, x:x+w]

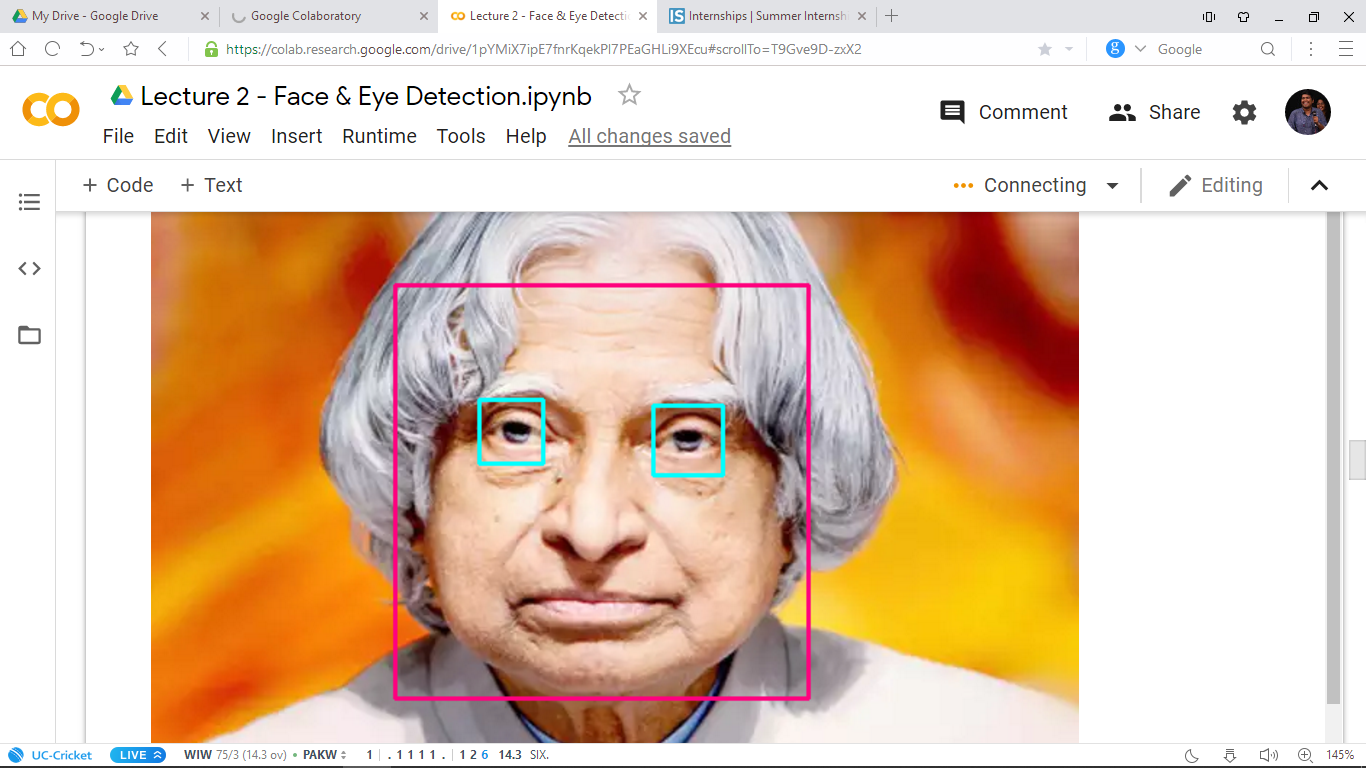
    eyes = eye\_classifier.detectMultiScale(roi\_gray,1.3,6)

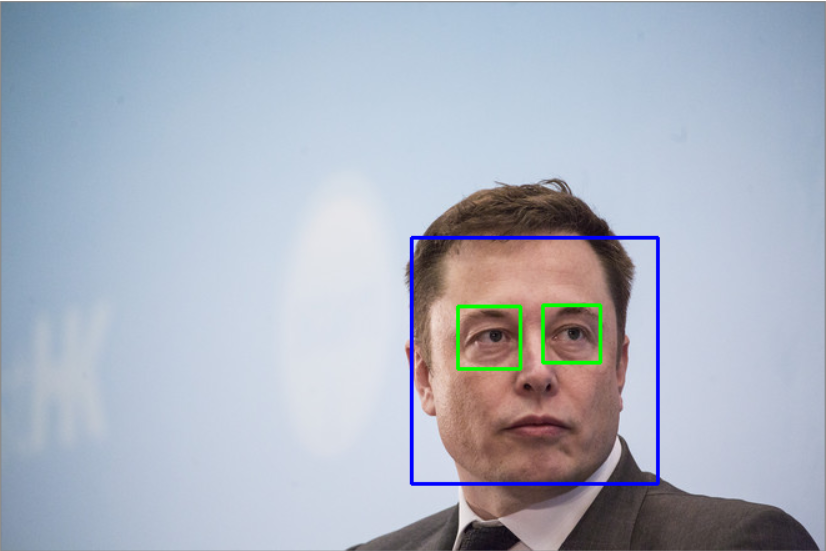
    for (ex,ey,ew,eh) in eyes:

        cv2.rectangle(roi\_color,(ex,ey),(ex+ew,ey+eh),(255,255,0),2)

        cv2\_imshow(img)

OUTPUT:





CAR AND PEDESTRIAN DETECTION USING HAAR CASCADES

**1.CAR DETECTION IN VIDEO**

import cv2

import time

import numpy as np

# Create our body classifier

car\_classifier = cv2.CascadeClassifier('/content/drive/My Drive/HAAR/Haarcascades/haarcascade\_car.xml')

# Initiate video capture for video file

cap = cv2.VideoCapture('/content/drive/My Drive/HAAR/image\_examples/cars.avi')

# Loop once video is successfully loaded

while cap.isOpened():

    time.sleep(.05)

    # Read first frame

    ret, frame = cap.read()

    gray = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)

    # Pass frame to our car classifier

    cars = car\_classifier.detectMultiScale(gray, 1.4, 2)

    # Extract bounding boxes for any bodies identified

    for (x,y,w,h) in cars:

        cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)

        cv2\_imshow( frame)

    if cv2.waitKey(1) == 13: #13 is the Enter Key

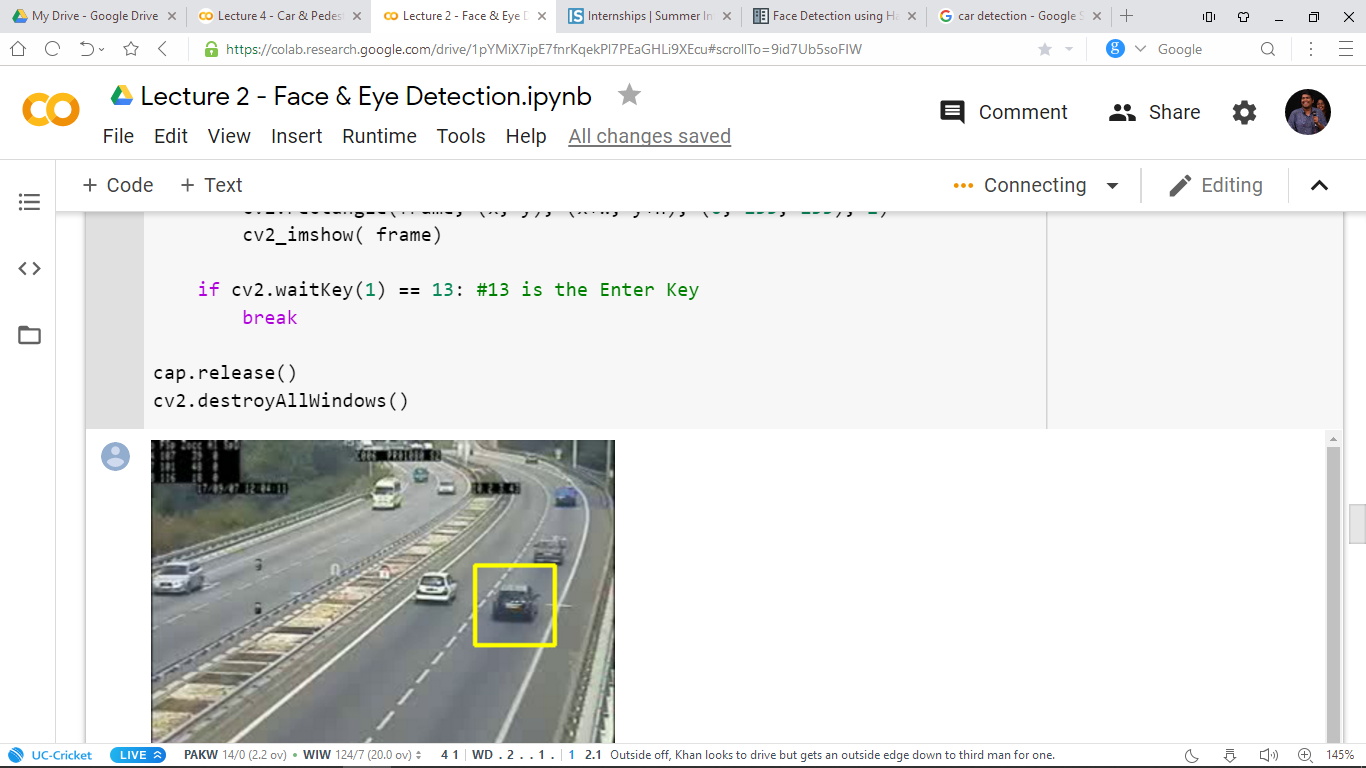
        break

cap.release()

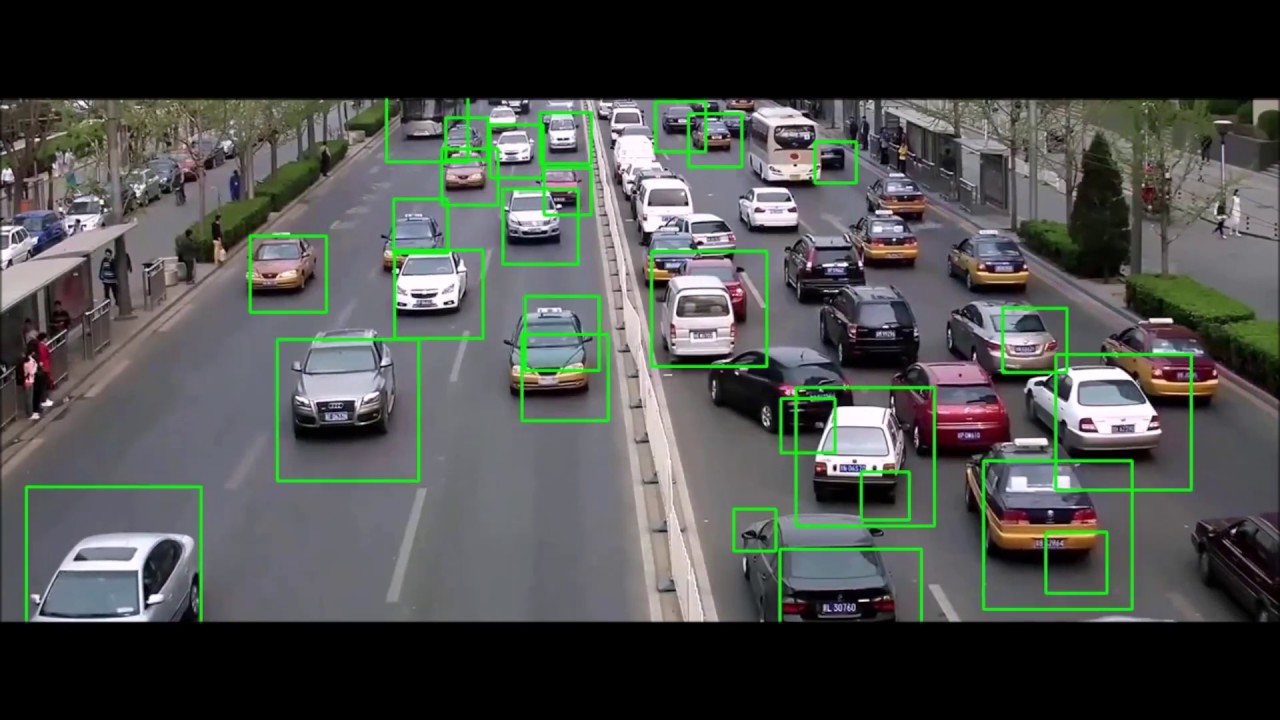
cv2.destroyAllWindows()

OUTPUT:

Car detection on video



Car detection on image:



PEDESTRIAN DETECTION

CODE:

import cv2

import numpy as np

# Create our body classifier

body\_classifier = cv2.CascadeClassifier('/content/drive/My Drive/HAAR/Haarcascades/haarcascade\_fullbody.xml')

# Initiate video capture for video file

cap = cv2.VideoCapture('/content/drive/My Drive/HAAR/image\_examples/walking.avi')

# Loop once video is successfully loaded

while cap.isOpened():

    # Read first frame

    ret, frame = cap.read()

    #frame = cv2.resize(frame, None,fx=0.5, fy=0.5, interpolation = cv2.INTER\_LINEAR)

    gray = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)

    # Pass frame to our body classifier

    bodies = body\_classifier.detectMultiScale(gray, 1.2, 3)

# Extract bounding boxes for any bodies identified

    for (x,y,w,h) in bodies:

        cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 255), 2)

        cv2\_imshow( frame)

    if cv2.waitKey(1) == 13: #13 is the Enter Key

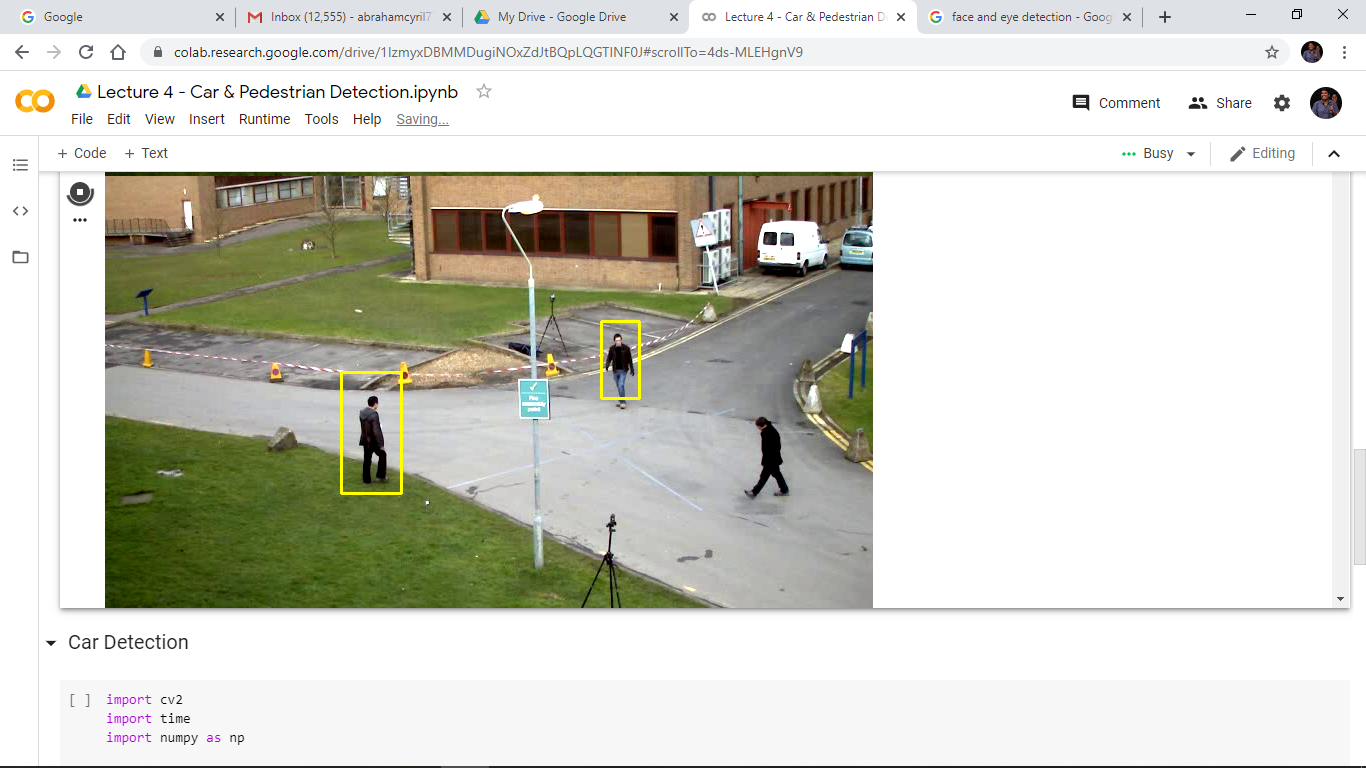
        break

cap.release()

cv2.destroyAllWindows()

OUTPUT

Pedestrian detection in video



Below pedestrian detection applied on image

