Capstone Project Report

Face Detection

Name: Keshav Anand

Course: AI and ML

(Batch-4)

Duration: 12 months

Problem Statement: Build a model which will detect the faces in the given image and also give the count of number of faces.

Prerequisites

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

Python 3.6 This setup requires that your machine has latest version of python. 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 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 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 below commands in command prompt/terminal to install these packages pip install -U scikit-learn pip install numpy pip install scipy if you have chosen to install anaconda then run below commands in anaconda prompt to install these packages conda install -c scikit-learn conda install -c anaconda numpy conda install -c anaconda scipy

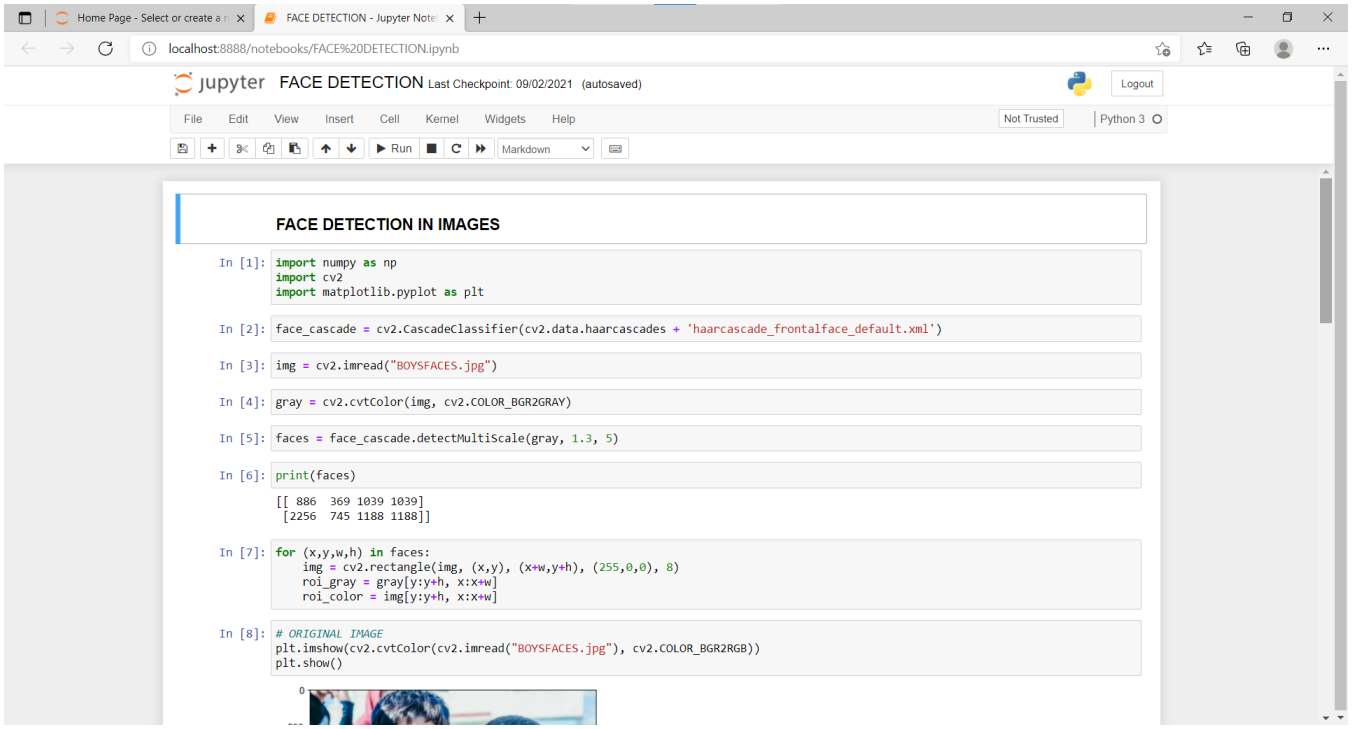
Dataset used :

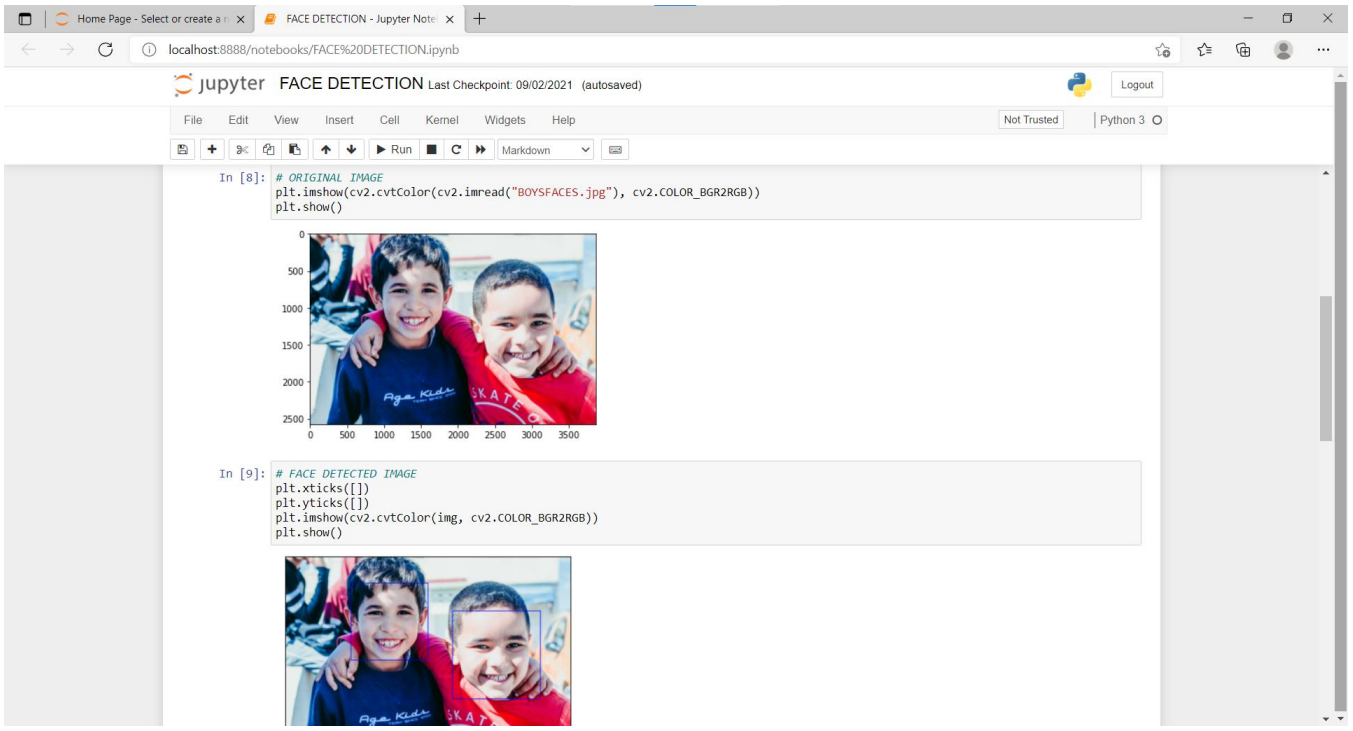
The Image used in this process is a random image downloaded from the internet.

Method used for Detection :

Haar-Cascade Classifier

ScreenShots Of Code and Output :





Home Page - Select or create a n | FACE DETECTION - Jupyter Note

localhost:8888/notebooks/FACE%20DETECTION.ipynb

jupyter **FACE DETECTION** Last Checkpoint: 09/02/2021 (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help | Not Trusted | Python 3

Run | Markdown

In [10]:
```
# FACE DETECTED WITH COUNT
text = "FACES: "+str(len(faces))
img = cv2.putText(img, text, (100,300), cv2.FONT_HERSHEY_PLAIN, 10, (0,0,255), 20)
plt.xticks([])
plt.yticks([])
plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
plt.show()
```

FACE DETECTION IN VIDEO

In [17]:
```
# create a viddeo capture object and read from input file
cap = cv2.VideoCapture("facemoving.mp4")

# face detection in video
face_cascade1 = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')

# read untile video is completed
while(True):
    # capture frame by frame
    ret, frame = cap.read()
```

Home Page - Select or create a n | FACE DETECTION - Jupyter Note

localhost:8888/notebooks/FACE%20DETECTION.ipynb

jupyter **FACE DETECTION** Last Checkpoint: 09/02/2021 (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help | Not Trusted | Python 3

Run | Markdown

FACE DETECTION IN VIDEO

In [17]:
```
# create a viddeo capture object and read from input file
cap = cv2.VideoCapture("facemoving.mp4")

# face detection in video
face_cascade1 = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')

# read untile video is completed
while(True):
    # capture frame by frame
    ret, frame = cap.read()

    # convert video into gray video without color
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    # detect faces in video
    faces = face_cascade1.detectMultiScale(gray, 1.3, 5)

    # draw rectangle boxes around the faces
    for (x,y,w,h) in faces:
        cv2.rectangle(frame, (x,y), (x+w,y+h), (255,0,0), 3)

    # display resulting frame
    cv2.imshow("FRAME", frame)

    # press q on keyboard to exit
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

# release video capture object
cap.release()

# close all frames
cv2.destroyAllWindows()
```