

FACIAL RECOGNITION AND TRACKING

A DAY WISE PROJECT REPORT

GIRL SCRIPT DEVELOPER TECH CAMP HACK-IN PROJECT

Hack –In is a week-long coding challenge in which the participants build a small-scale project using new technology.

BY - SIMAR PREET SINGH

Guru Teg Bahadur Institute Of Technology, Rajauri
Garden , Delhi

Github=simarpreetsingh019

ACKNOWLEDGEMENT



I would like to express my special thanks of gratitude to my mentor Maitree Rawat and team of GirlScript Jaipur who gave me the golden opportunity to do this wonderful project on Face Recognition, which also helped me in learning a lot of new things and Technologies. Moreover, it helped me to experience essence of open source. Also, I would like to thank all the co mentees for helping me to work on project.

ABSTRACT

The growing interest in computer vision of the past decade. Fuelled by the steady doubling rate of computing power every 13 months, face detection and recognition has transcended from an esoteric to a popular area of research in computer vision and one of the better and successful applications of image analysis and algorithm based understanding. Because of the intrinsic nature of the problem, computer vision is not only a computer science area of research, but also the object of neuro-scientific and psychological studies, mainly because of the general opinion that advances in computer image processing and understanding research will provide insights into how our brain work and vice versa. Because of general curiosity and interest in the matter, the author has proposed to create an application that would allow user access to a particular machine based on an in-depth analysis of a person's facial features.

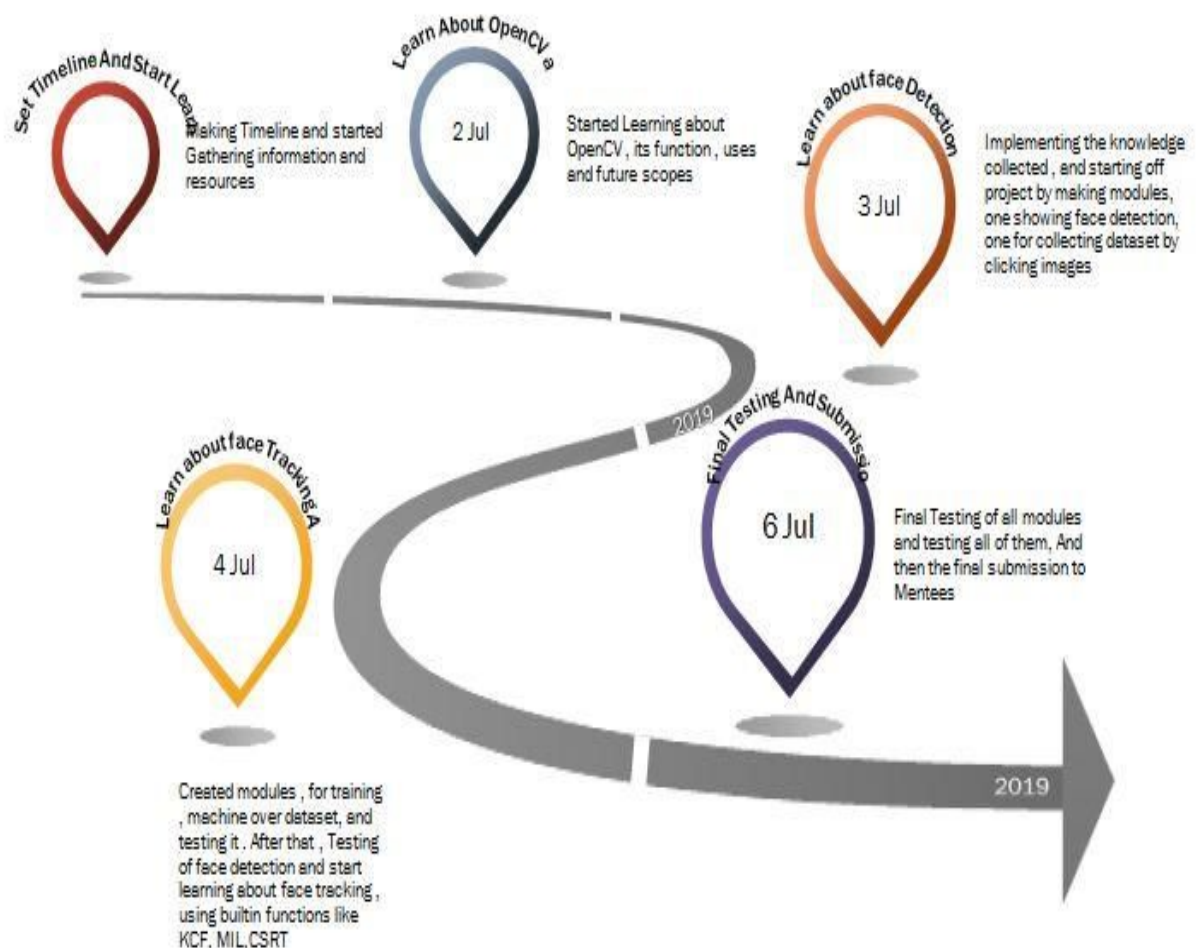
TECHNOLOGY STACK

1. OPENCV 4
2. PYTHON 3
3. PYCHARM IDE
4. WINDOW 10 OS
5. PIP
6. IMUTILS
7. NUMPY
8. PILLOW

DAY 1: CREATING TIMELINE

❖ Objective: To create timeline , to be followed during the week.

❖ Timeline Image:



DAY 2: LEARN ABOUT OPENCV

❖ Objective: Start Learning About OpenCV

❖ Code :

```
import numpy as np
import cv2

Facedetect=cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
eyedetect = cv2.CascadeClassifier('haarcascade_eye.xml')

cam = cv2.VideoCapture(0)

while(True):
    ret,img = cam.read()
    gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
    faces = facedetect.detectMultiScale(gray,1.3,5)
    for (x,y,w,h) in faces:
        cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2)
        roi_gray = gray[y:y+h, x:x+w]
        roi_color = img[y:y+h, x:x+w]

        eyes = eyedetect.detectMultiScale(roi_gray)
        for (ex,ey,ew,eh) in eyes:
            cv2.rectangle(roi_color,(ex,ey),(ex+ew,ey+eh),(0,255,0),2)

    cv2.imshow("face",img)
    if(cv2.waitKey(1)==ord('q')):
        Break

cam.release()
cv2.destroyAllWindows()
```

❖ Sample Output Screenshots:

DAY 3: LEARN ABOUT FACE

DETECTION

❖ Objective: Implementing knowledge gathered over day

❖ Code :

```
import numpy as np
import cv2

facedetect =
cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

sampleNum = 0

uid = int(input('enter user id'))

cam = cv2.VideoCapture(0)

while(True):
    ret,img = cam.read()
    gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
    faces = facedetect.detectMultiScale(gray,1.3,5)

    for(x,y,w,h) in faces:
        sampleNum+=1

cv2.imwrite('datasets/'+str(uid)+'_'+str(sampleNum)+'.jpg',gray[y:y+h,x:x+w])
        cv2.rectangle(img,(x,y),(x+w,y+h),(0,255,0),2)
        cv2.waitKey(100)
    cv2.imshow('face',img)
    cv2.waitKey(1)
    if(sampleNum>50):
        Break

cam.release()
cv2.destroyAllWindows()
```

❖ Sample Output Screenshots:

DAY 4: LEARNING TRACKERS

- ❖ Objective: Learning about trackers
- ❖ Description: during this time , I gathered knowledge about hoe trackers work, there principles, and how different trackers differentiate from each other.

Some of them were : LBPH

MIL

KCF

- ❖ Code : face and eye detector:

```
import numpy as np
import cv2

facedetect =
cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
eyedetect = cv2.CascadeClassifier('haarcascade_eye.xml')

sampleNum = 0

uid = int(input('enter user id'))
name = input('Enter your name')

cam = cv2.VideoCapture(0)

while(True):
    ret,img = cam.read()
    gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
    faces = facedetect.detectMultiScale(gray,1.3,5)

    for(x,y,w,h) in faces:
```



```

        sampleNum+=1
        cv2.imwrite('datasets
color/data/'+str(uid)+'_'+str(sampleNum)+'.jpg',gray[y:y+h,x:x+w])
        cv2.rectangle(img,(x,y),(x+w,y+h),(0,255,0),2)
        roi_gray = gray[y:y+h, x:x+w]
        roi_color = img[y:y+h, x:x+w]

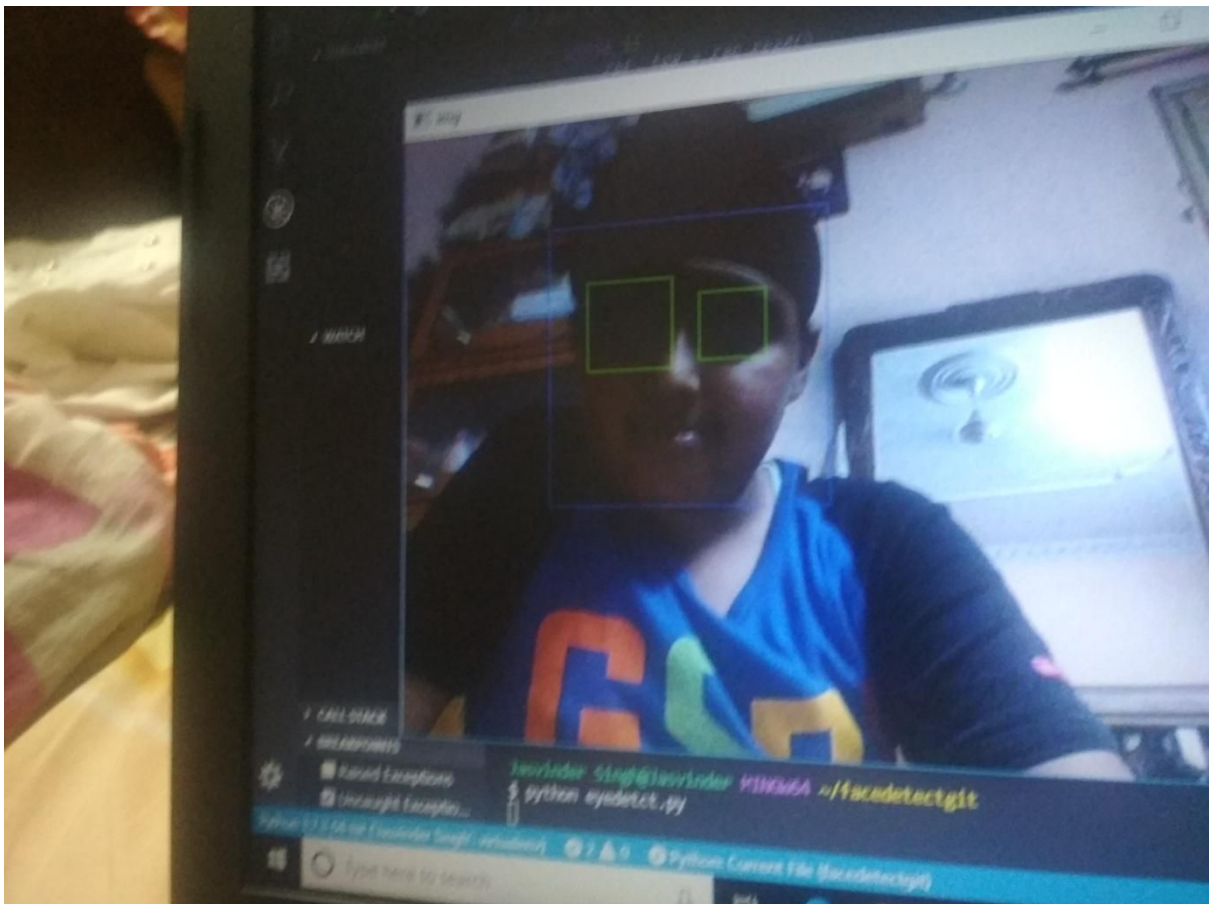
        eyes = eyedetect.detectMultiScale(roi_gray)
        for (ex,ey,ew,eh) in eyes:
            cv2.rectangle(roi_color,(ex,ey),(ex+ew,ey+eh),(0,255,0),2)
            cv2.waitKey(500)

        cv2.imshow('face',img)
        cv2.imshow('eye',img)
        cv2.waitKey(1)
        if(sampleNum>10):
            Break

cam.release()
cv2.destroyAllWindows()

```

❖ Sample Output Screenshots:



How face and eye detector were working on showing face

DAY 5: DATASET COLLECTION AND TRAINING

- ❖ Objective: Dataset collection and Training
- ❖ Images if any:
- ❖ Code if any:

```
import numpy as np
import cv2

facedetect =
cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

sampleNum = 0

uid = int(input('enter user id'))

cam = cv2.VideoCapture(0)

while(True):
    ret,img = cam.read()
    gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
    faces = facedetect.detectMultiScale(gray,1.3,5)

    for(x,y,w,h) in faces:
        sampleNum+=1

cv2.imwrite('datasets/'+str(uid)+'_'+str(sampleNum)+'.jpg',gray[y:y+h,x:x+w])
        cv2.rectangle(img,(x,y),(x+w,y+h),(0,255,0),2)
        cv2.waitKey(100)
    cv2.imshow('face',img)
    cv2.waitKey(1)
    if(sampleNum>50):
        Break

cam.release()
cv2.destroyAllWindows()
```

Training over the dataset

```
import numpy as np
import os
from PIL import Image
import cv2

recog = cv2.face.LBPHFaceRecognizer_create()
path = 'datasets'

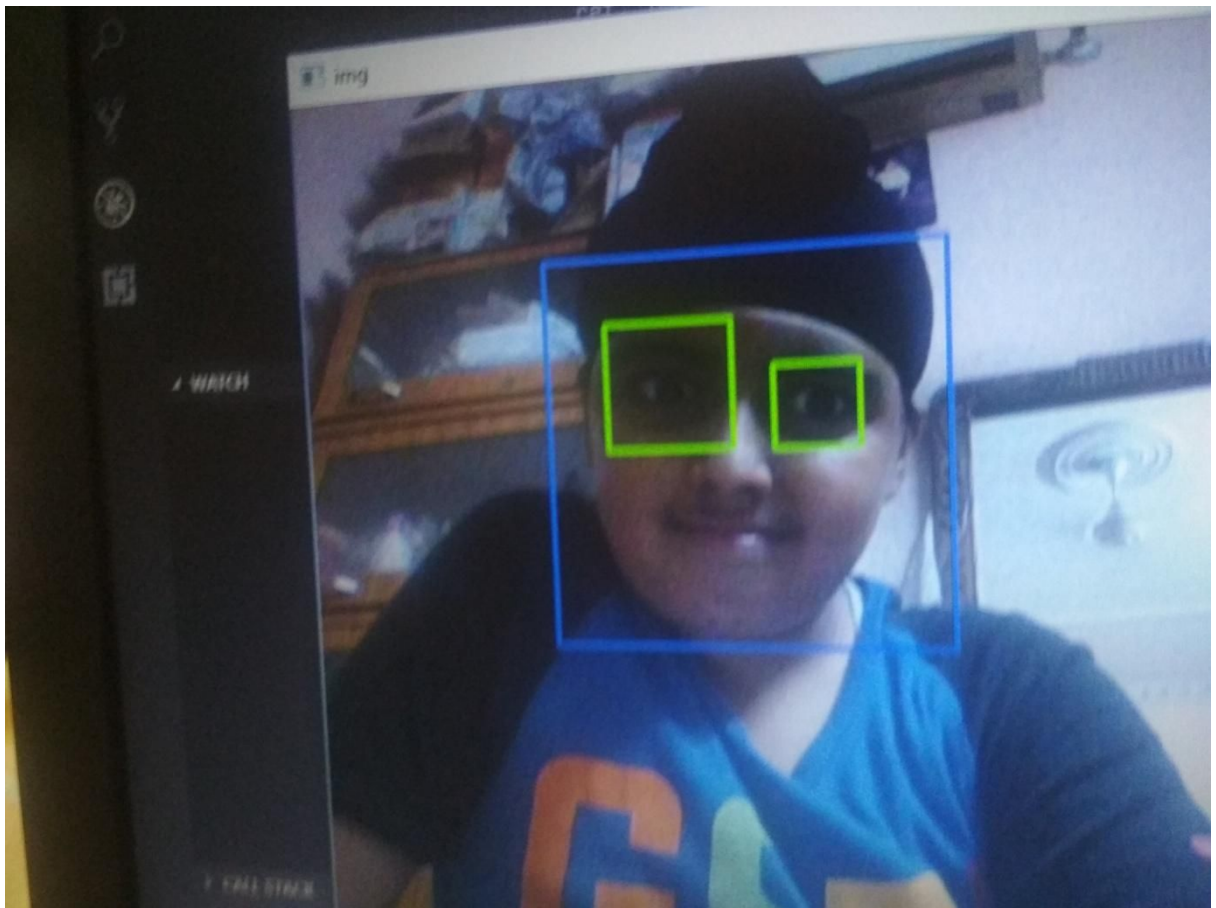
def getImageswithID(path):
    imagePaths = [os.path.join(path,f) for f in os.listdir(path)]
    faces = []
    IDs = []

    for imgpath in imagePaths:
        faceImg = Image.open(imgpath)
        faceNp = np.array(faceImg,'uint8')
        print(os.path.split(imgpath)[-1].split('_')[0])
        ID = int(os.path.split(imgpath)[-1].split('_')[0])
        faces.append(faceNp)
        IDs.append(ID)
        cv2.imshow('training',faceNp)
        cv2.waitKey(10)
    return IDs, faces

IDs,faces = getImageswithID(path)
recog.train(faces,np.array(IDs))
recog.write('trainingData.yml')
cv2.destroyAllWindows()
```

DAY 6: TESTING DATA

- ❖ Objective: Testing Data And Finalising it
- ❖ Images if any:



❖ Code

#Testing Over Data

```
import numpy as np
import cv2

facedetect =
cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
rec = cv2.face.LBPHFaceRecognizer_create()
rec.read('trainingData.yml')

id = 0
fontFace = cv2.FONT_HERSHEY_SIMPLEX
fontScale = 1
fontColor = (0,0,255)

id_map = ['Simar','Siddhart']

cam = cv2.VideoCapture(0)

while(True):
    ret,img = cam.read()
    gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
    faces = facedetect.detectMultiScale(gray,1.3,5)

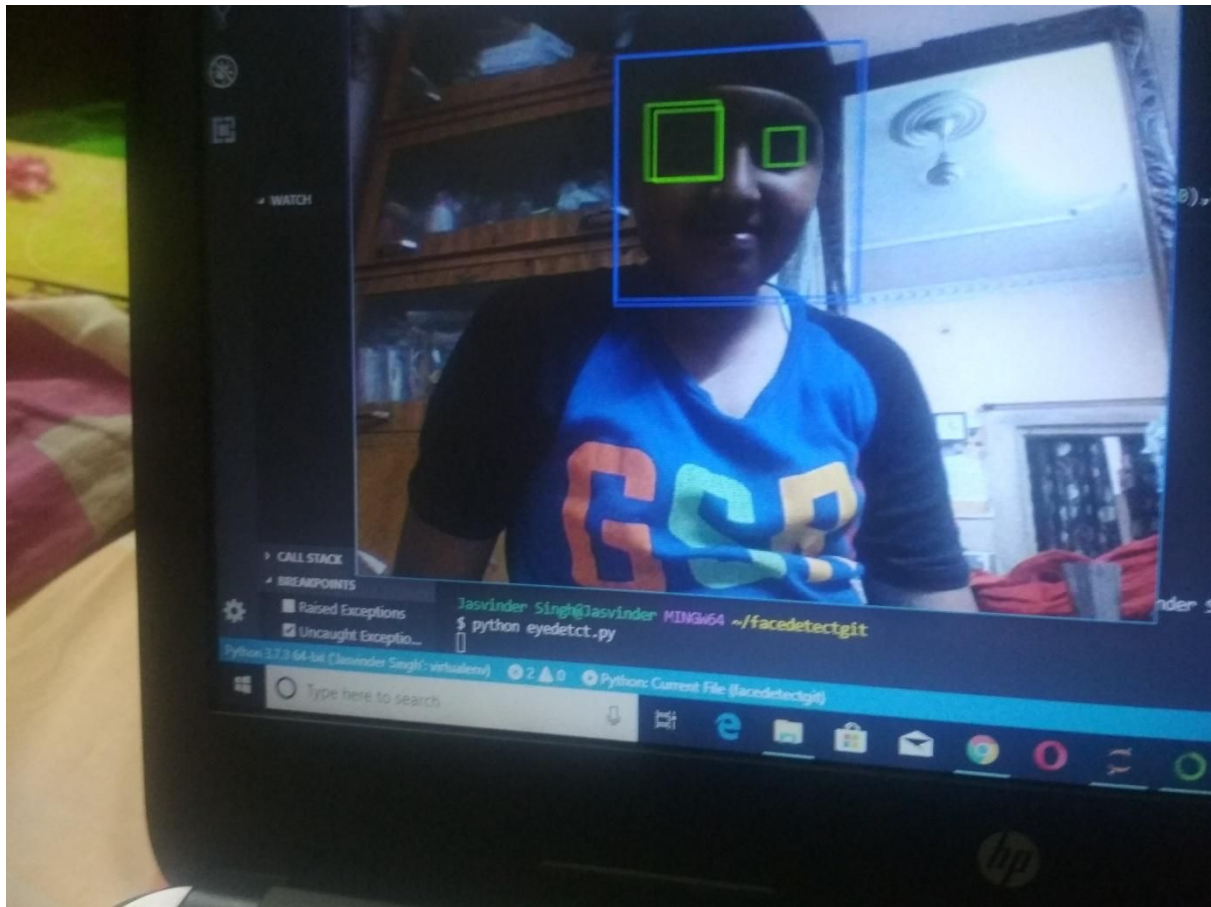
    for(x,y,w,h) in faces:
        cv2.rectangle(img,(x,y),(x+w,y+h),(0,255,0),2)
        id,conf = rec.predict(gray[y:y+h,x:x+w])

cv2.putText(img,str(id_map[id-1])+'_'+str(conf),(x,y+h),fontFace,fontS
cale,fontColor)

    cv2.imshow("face",img)
    if(cv2.waitKey(1)==ord('q')):
        break

cam.release()
cv2.destroyAllWindows()
```

❖ Sample Output Screenshots:



FUTURE SCOPE

IT CAN FURTHER BE USED IN MANY WAYS, DETECTION OF FACES AND PREDICTING THEIR EMOTIONS, OR JUST LIKE SNAPCHAT ADDING OTHER FILTERS, OR USED FOR ATTENDENCE IN SCHOOLS AND MANY MORE THINGS

REFERENCES

OPENCV DOCUMENTS

W3SCHOOLS

STACKOVER FLOW

GEEKS FOR GEEK

SUPERDATASCIENCE.COM

PLURALSIGHT.COM

KAGGLE