

PRODDEC PYTHON DAY 10

Detect face from the image

STEPS:

1.Download haarcascade file for frontal face from the opency github page.

https://github.com/opencv/opencv/tree/master/data/haarcascades

2.Download hd images image for face detection.

3.Run the code and enjoy.

Additional link 1: - face detection using opency

https://www.datacamp.com/community/tutorials/face-detection-python-opencv

https://www.pythonpool.com/face-detection-using-opencv-and-python/

Additional link 2: - name the certificate application.

https://www.youtube.com/channel/UCvzLCuUAHHeTpGhc1Vewsuw

https://drive.google.com/file/d/1IBUTAgzINew5bMKXy_54rg2IQtSsKPT5/view

Code for face detection from image

```
In [13]:
         import cv2
         # Reading cricketer image.
         img = cv2.imread("cricketer.jpg")
         # Converting to gray
         gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
         # Initialize the face recognizer (default face haar cascade)
         face_cascade = cv2.CascadeClassifier("haarcascade_fontalface_default.xml")
         # Detect all the faces in the image
         faces = face_cascade.detectMultiScale(gray, 1.3, 5)
         # For every face, draw a green rectangle
         for x, y, w, h in faces:
             cv2.rectangle(img,(x, y),(x + w, y + h),(0, 255,0),thickness=2)
             cv2.putText(img, "people", (x,y-10), cv2.FONT_HERSHEY_SIMPLEX,1, (0,255,0),2)
         cv2.imwrite("detection-image.jpg",img)
         cv2.imshow("detection image", img)
         cv2.waitKey(0)
         cv2.destroyWindow("detection image")
```





camera/video

```
In [18]: import cv2
                                # Create a new cam object
                                cap = cv2.VideoCapture(0)
                                # Initialize the face recognizer (default face haar cascade)
                                face_cascade = cv2.CascadeClassifier("haarcascade_fontalface_default.xml")
                                while(cap.isOpened()):
                                             # Read the image from the cam
                                             ret, image = cap.read()
                                             if ret==True:
                                                          # Converting to grayscale
                                                          image_gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
                                                          # Detect all the faces in the image
                                                          faces = face_cascade.detectMultiScale(image_gray, 1.3, 5)
                                                          # For every face, draw a red rectangle.
                                                          for x, y, w, h in faces:
                                                                        cv2.rectangle(image, (x, y), (x + w, y + h), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), (0, 0, 255), 
                                                          cv2.imshow("video frame", image)
                                                          # If q button is pressed, quit.
                                                          if cv2.waitKey(1) == ord("q"):
                                                                        break
                                             else:
                                                          break
                                cap.release()
                                cv2.destroyAllWindows()
```

```
In [ ]:

In [ ]:
```