

Project:- Live Face Detection Using OpenCV

Team Name: Tech_blasters



Problem Statement:

A general statement of the face detection problem (in computer vision) can be formulated as follows: given still or video images of a scene, detect one or more persons. Face Detection where a photo is searched to find a face, then the image is processed to crop and extract the person's face for easier recognition.

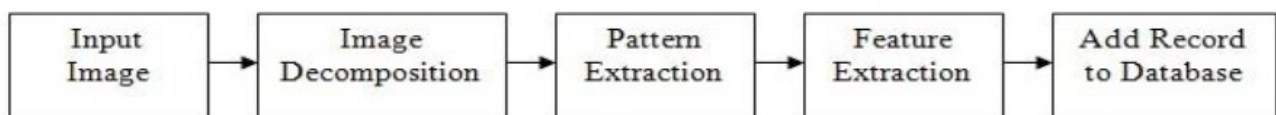
Proposed Solution:

OpenCV uses a type of face detector called a Haar Cascade classifier. Given an image, which can come from a file or from live video, the face detector examines each image location and classifies it as "Face" or "Not Face." It detects the face and draws a rectangle around the face.

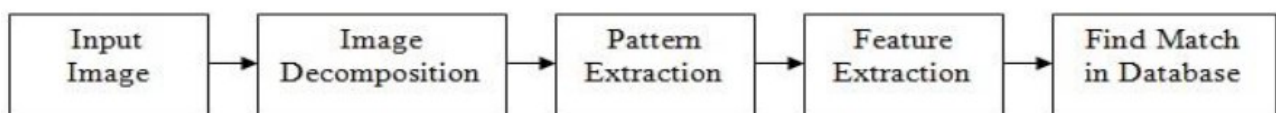
Software Requirements:

- Open CV
- Haarcascade Classifier
- Python

Block Diagram And Discription:



Procedure for Facial Feature's Extraction.



Procedure for Face Recognition.

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 used that model to detect faces in live .

Steps:

Step 1:-import modules.

Step2:-code to capture the live photo/video

Step3:-create a model using haarcascade_frontal_face_default.xml” using cascadeClassifier.

Step4:-it detects the live faces.

Step5:-draw the rectangle on live detected face.

Step6:-Displays the live face detected photo/video.

Code:

```
In [ ]: import cv2

In [ ]: cap=cv2.VideoCapture(0)

In [ ]: ret,photo=cap.read()

In [ ]: model=cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

# FOR FACE DETECTION IN LIVE CAPTURED IMAGE ##

In [ ]: x=model.detectMultiScale(photo)

In [ ]: nphoto=cv2.rectangle(photo,(x[0][0],x[0][1]),(x[0][0]+x[0][2],x[0][1]+x[0][3]),[255,0,0],3)

In [ ]: cv2.imshow("hi",nphoto)
cv2.waitKey()==13

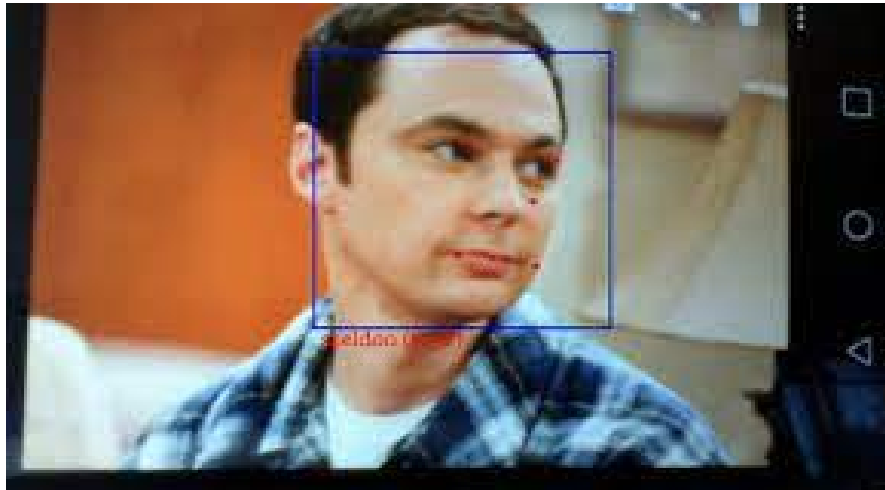
cv2.destroyAllWindows()

In [ ]: cap.release()

# FOR FACE DETECTION IN LIVE CAPTURED VIDEO ##

In [ ]: while True:
    ret,photo=cap.read()
    x=model.detectMultiScale(photo)
    print(x)
    nphoto=cv2.rectangle(photo,(x[0][0],x[0][1]),(x[0][0]+x[0][2],x[0][1]+x[0][3]),[255,0,0],3)
    x1=x[0][0]
    y1=x[0][1]
    x2=x[0][0]+x[0][2]
    y2=x[0][1]+x[0][3]
    bphoto=photo[x1:x2,y1:y2]
    b=cv2.blur(bphoto,(25,25))
    photo[x1:x2,y1:y2]=b
    cv2.imshow("hi",photo)
    if cv2.waitKey(10)==13:
        break
    cv2.destroyAllWindows()
```

OUTPUT:



Conclusion:

You can usually improve the face detection accuracy by using more input images, atleast 50 per person, by taking more photos of each person, particularly from different angles and lighting conditions. You can easily identify the faces from existing photos. face detection is relatively easy to do in real-time. it is so useful in security cameras, traffic areas and public places.

Team Member Details:

S.NO	NAME	ID NO.	MOBILE NUMBER	EMAIL
1	P.NARMADA	O170115	7095986464	o170115@rguktong.ac.in
2	CH.NAGAMANI	O170055	8688892998	chukkabotlanagamani@gmail.com
3	G.UMA DURGA	O170098	9491269577	umadurga1610@gmail.com
4	K.ANUDEEPTHI	O170166	6303484114	o170166@rguktong.ac.in
5	P.SAI LAKSHMI	O170593	9398293543	o170593@rguktong.ac.in