Project: Auto-capture Selfie by Detecting Smile

OpenCV is an open-source library for computer vision, with a focus on real-time applications. It focuses mainly on video capture/processing, image processing, and analysis (like face and object detection). It has many built-in functions and pre-trained models.

Haar Cascade is an ML object detection algorithm used to identify objects in an image (treated as a matrix i.e. 2D grid here) or video.

a cascade function is trained from a lot of positive and negative images which is then used to detect objects in other images. It can be trained to identify almost any object. In this project, we will be using these pre-trained files.

The algorithm has four steps:

1. Haar Feature Selection
2. Creating Integral Images
3. Adaboost Training
4. Cascading Classifiers.

Steps Involved to implement Smile Detection and Selfie Capture Project

1. We first import the openCV library.
2. Now start webcam in the second line using the VideoCapture function of cv2.
3. Then, include haarcascade files in the python file.
4. Video is nothing but a series of images so we will run an infinite while loop for the same.
5. Then we are reading images from the video through read().
6. As feature recognition is more accurate in gray images we will convert the image to gray image using cvtColor() and BGR2GRAY which are basic openCV functions.
7. Now we will read faces using an already included haarcascade file and detectMultiscale() function where we pass gray image, ScaleFactor, and minNeighbors.

- **ScaleFactor:** Parameter specifying zoom image, accuracy depends on it so we will keep it close to 1 but not very close as if we take 1.001(very close to 1), then it would detect even shadows so 1.1 is good enough for the face.
- **minNeighbors:** Parameter specifying how many neighbors each rectangle should have to retain it.

8. If it detects a face we will draw an outer boundary of the face using rectangle() method of cv2 containing 5 arguments: image, initial point (x, y), an endpoint of principal diagonal (x + width, y + height), color of the rectangular periphery and last parameter is the thickness of drawn rectangular periphery.
9. If the face is detected then we will similarly detect a smile and if a smile is detected too we will print Image<cnt> saved in the cmd/terminal and then we have to provide the location of the folder in which we want to save the images.
10. To save the images we will use imwrite() which takes 2 parameters- location and image.
11. To prevent memory overflow we will just save 2 images in one run and thus useif statement which breaks the loop if cnt>=2.
12. To break infinite loop, we have used an if statement which becomes true when we press ‘q’ denoting ‘quit’.
13. At last, we will release the video.
14. Do not forget to destroy all the windows.

Python code:

Importing libraries:

```
import cv2
```

Importing cascade:

```
video = cv2.VideoCapture(0)
faceCascade = cv2.CascadeClassifier("C:/Users/GOPINADH/Desktop/datasets/dataset/haarcascade_frontalface_default.xml")
smileCascade = cv2.CascadeClassifier("C:/Users/GOPINADH/Desktop/datasets/dataset/haarcascade_smile.xml")
```

Auto Capturing

```
while True:
    success,img = video.read()
    grayImg = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    faces = faceCascade.detectMultiScale(grayImg,1.1,4)
    cnt=500
    keyPressed = cv2.waitKey(1)
    for x,y,w,h in faces:
        img = cv2.rectangle(img,(x,y),(x+w,y+h),(0,0,0),3)
        smiles = smileCascade.detectMultiScale(grayImg,1.8,15)
        for x,y,w,h in smiles:
            img = cv2.rectangle(img,(x,y),(x+w,y+h),(100,100,100),5)
            print("Image "+str(cnt)+"Saved")
            path=r'C:\Users\GOPINADH\Pictures\images'+str(cnt)+'.jpg'
            cv2.imwrite(path,img)
            cnt +=1
            if(cnt>=503):

                break

    cv2.imshow('live video',img)
    if(keyPressed & 0xFF==ord('q')):
        break
```

Auto image save:

```
Image 500Saved
Image 500Saved
Image 500Saved
Image 500Saved
Image 500Saved
Image 500Saved
Image 501Saved
Image 500Saved
Image 500Saved
Image 500Saved
Image 500Saved
Image 500Saved
Image 500Saved
Image 500Saved
Image 500Saved
Image 500Saved
Image 500Saved
Image 500Saved
Image 500Saved
Image 501Saved
```

Release video:

```
video.release()
cv2.destroyAllWindows()
```