ABSTRACT

Face detection technology has widely attracted attention due to its enormous application value and market potential, such as face recognition and video surveillance system. Real-time face detection not only is one part of the automatic face recognition system but also is developing an independent research subject. So, there are many approaches to solve face detection. Here the modified AdaBoost algorithm based on OpenCV is presented, and experiments of real-time face detecting are also given. The result shows that the method of face detection with AdaBoost algorithm is simpler, smoother and more precise.

Introduction

In the fast-developing world, everything is getting digitalized, so is the security also, everyone wants their work to be done fast and simple. Face Detection has been one of the hottest topics of computer vision for the past few years. This technology has been available for some years now and is being used all over the place. Currently few of the industries and banks have installed such machines for security purposes. Even few of the mobiles right now use this technique, so that only their owners will be able to access their mobile.

The face detection process is one which take the image of face and divides it into various pixels, convert it into one dimensional array and compares with the one stored in the database. So, this is how it works when you look at image processing device it extracts your face from an image and then it compares with the one it saved and checks if they both match and if they do it will unlock itself. And this all happens in blink of a second.

The face detection process needs a high-quality camera with a information database, and a software to match the recorded faces with the images in the database. For extracting image features we use Classifiers which are already defined in the modules with help of these classifiers we are going to detect face using detectMultiscale feature.

Working

Machine learning is the ability for computers to learn and act without being explicitly programmed. At the core of every machine learning algorithm, all the algorithm is trying to do is find a relationship from the input X to the output Y. How exactly it finds this function varies by the algorithm. But almost all algorithms require a lot of computations. And a lot of revisions and usually a good amount of data. In this first we are going to input maximum amount of data i.e. here number of images of faces and we wanted to train the dataset according to our algorithm.

Dataset creator

In this first we are created a dataset folder in which we are going to capture images and save them into this folder. This process is called dataset generator. We are importing two modules called cv2 and numpy.

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

this above shown code detect the faces using CascadeClassifier algorithm which use xml file as input in which many different orientations of faces were already defined based on this it will detect the faces. As this algorithm is free for student purpose in this we can detect only 90-degree alignment of faces only.

cam= cv2.VideoCapture(0)

this will capture the video using web cam which is defined as 0.

faces=faceDetect.detectMultiScale(gray,1.3,5)

this will draw the rectangular box on the detected face where gray is the grayscale image and 1.3 is the scaleFactor and 5 is the minNeighbors. By using loop, we are continuously detecting the faces and drawing a rectangle over it.

Training Dataset

Next we are going to train these captured images using createLBPHFaceRecognizer () algorithm. This trains the faces according to the inputted id number.

from PIL import Image

this is imported because to read image files from HDD and convert them into pixels which further one dimensional array. We are converting into 1D array because comparing is faster in 1D array compared to multi-dimensional array. After training we are getting output as array of numbers which corresponds to a particular image. This one we are saving as trainningData.yml in training folder. Yml extension because this is very fast to read by computers.

recognizer=cv2.createLBPHFaceRecognizer();

Detecting Real Time

In next step we are going to detect images in real time.in this we are capturing the images via webcam continuously in real time.the captured images are compared with the dataset i.e. data base which is stored as yml file the read data are compared with this array of set and based on id number provide we are going to detect image with their details provided during data set creation.

```
faceDetect=cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
rec=cv2.createLBPHFaceRecognizer();
```

Using above algorithms, we are going to detect the faces and drawing a rectangular box on faces edges. The image which is captured is compared with the array present in trainningData.yml file. Which predicts the face using inbuilt module predict.

```
id,conf=rec.predict(gray[y:y+h,x:x+w])
```

where id is the id number inputted to particular person and inputting a gray scaled image within the limits of rectangular box. This will predict the face of person with the data base present if not present it will show not in database.

Code

```
######   Dataset Creator  #######


import cv2
import numpy as np


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


cam= cv2.VideoCapture(0)


id= input('Enter User id : ')
sample_num=0


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


    for (x,y,w,h)in faces :
        sample_num +=1
        cv2.imwrite("dataset\\user" +'.'+ str(id)+'.'+ str(sample_num)+'.jpg',gray[y:y+h,x:x+w])
        cv2.rectangle(img,(x,y),(x+w,y+h),(0,0,255),2)
        cv2.waitKey(100)


    cv2.imshow("Face",img)
    cv2.waitKey(1)
    if(sample_num > 49):
        break
```

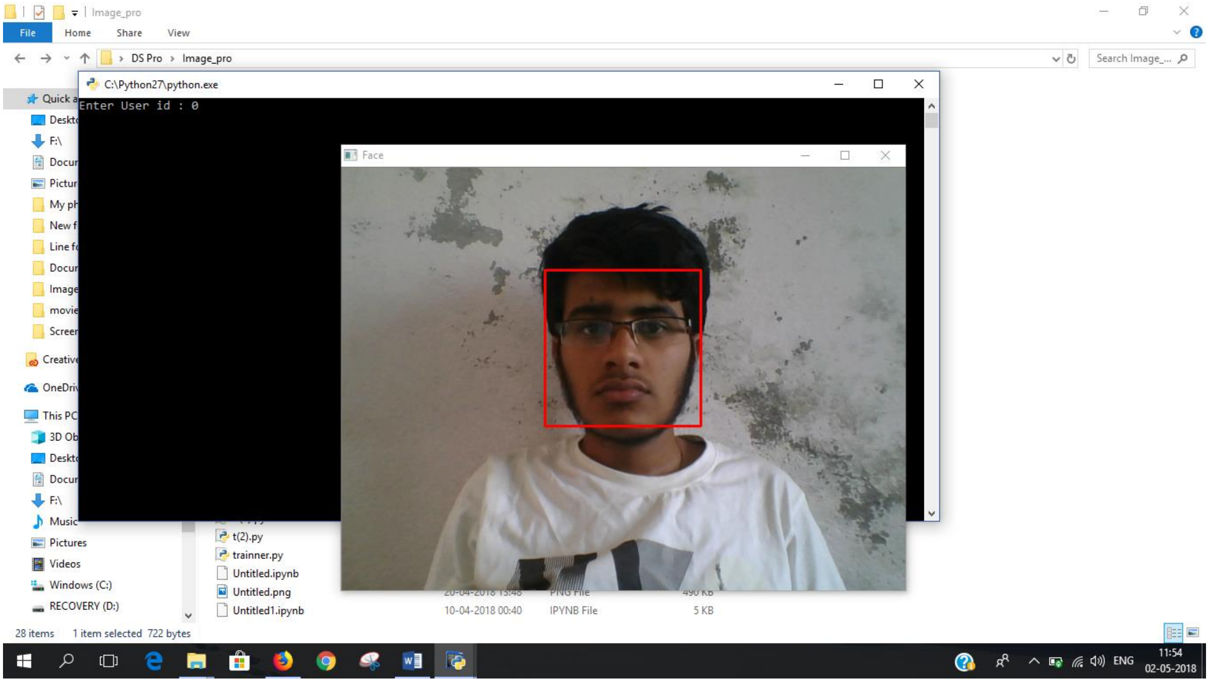
```
cam.release()
cv2.destroyAllWindows()



##### Training Dataset  #######


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


recognizer=cv2.createLBPHFaceRecognizer();
path='dataSet/user'


def getImagesWithID(path):
        imagePaths=[os.path.join(path,f) for f in os.listdir(path)]
        faces=[]
        IDs=[]
        for imagePath in imagePaths:
                faceImg=Image.open(imagePath).convert('L');
                faceNp=np.array(faceImg,'uint8')
                ID=int(os.path.split(imagePath)[-1].split('.')[1])
                faces.append(faceNp)
                print ID
                IDs.append(ID)
                cv2.imshow("training",faceNp)
                cv2.waitKey(10)
        return IDs,faces


Ids,faces=getImagesWithID(path)
```

```
recognizer.train(faces,np.array(Ids))
recognizer.save('recognizer/trainningData.yml')
cv2.destroyAllWindows()



####### Detecting Real Time ######


import cv2
import numpy as np
faceDetect=cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
cam=cv2.VideoCapture(0);
rec=cv2.createLBPHFaceRecognizer();
rec.load("recognizer/trainningData.yml")
id=0
ch='Not in Database..!!'
font=cv2.cv.InitFont(cv2.cv.CV_FONT_HERSHEY_COMPLEX_SMALL,1,1,0,2)
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),(0,0,255),2)
        id,conf=rec.predict(gray[y:y+h,x:x+w])
        if(id==1):
            ch="Abhishek S Patil"
            cv2.cv.PutText(cv2.cv.fromarray(img),str('SR NO. :
160045'),(x,y+h+20),font,255);
        elif(id==2):
            ch="Varshith A N"
            cv2.cv.PutText(cv2.cv.fromarray(img),str('SR NO. :
162017'),(x,y+h+20),font,255);
```

```
            elif(id==3):
                ch="Pavan"
                cv2.cv.PutText(cv2.cv.fromarray(img),str('SR NO. :
160092'),(x,y+h+20),font,255);
            cv2.cv.PutText(cv2.cv.fromarray(img),str(ch),(x,y+h),font,255);
            cv2.cv.PutText(cv2.cv.fromarray(img),str('B Sec ECE
Dept.'),(x,y+h+40),font,255);


      cv2.imshow("Face",img);
      if(cv2.waitKey(1)==ord('q')):
            break;
cam.release()
cv2.destroyAllWindows()
```

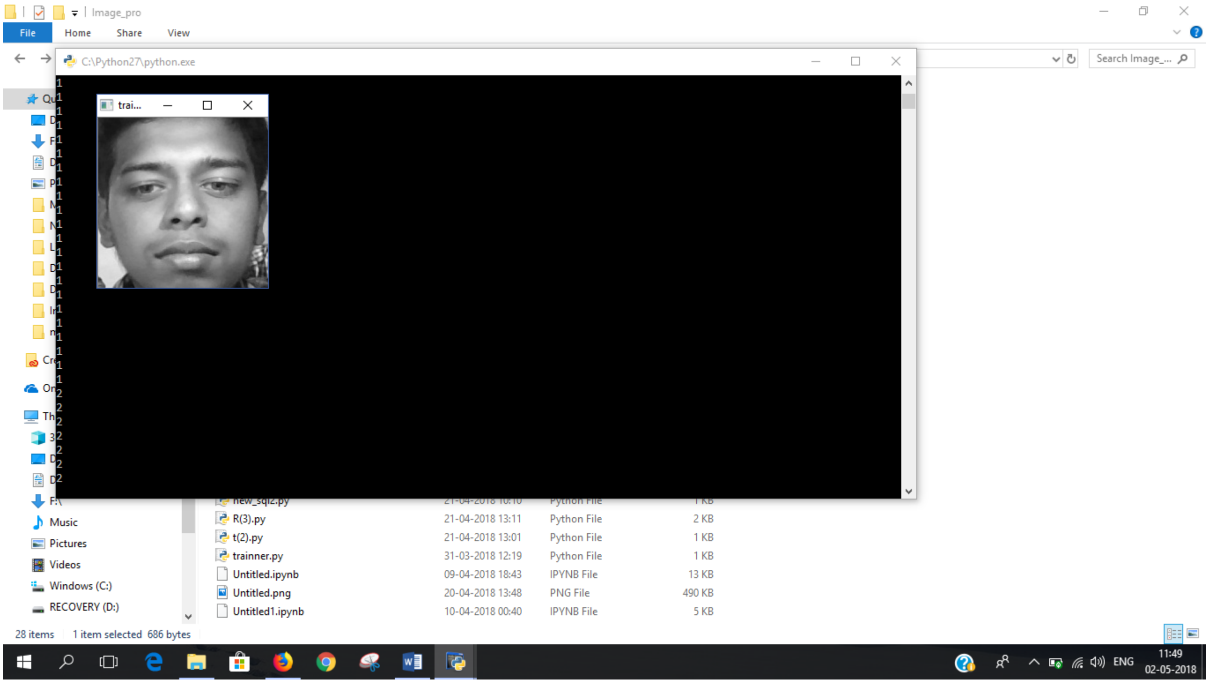
Snapshots

1] Dataset Creator



```
Enter User id : 0
```

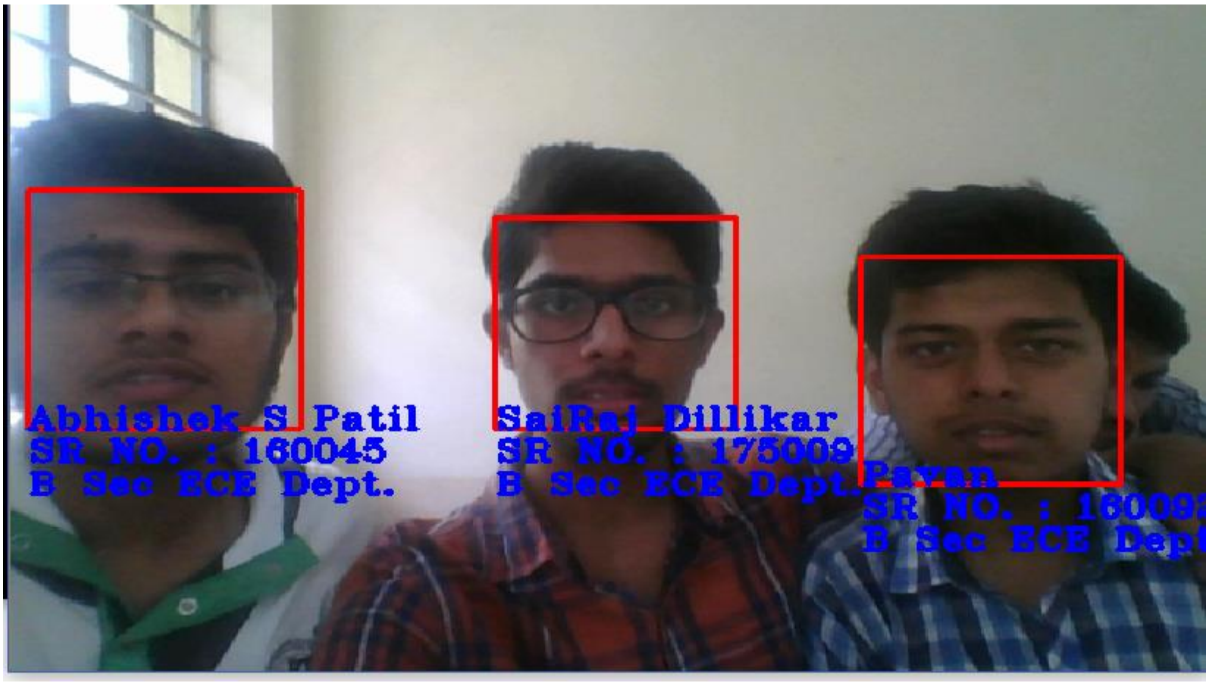
2] Tanning Dataset



```
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
2
2
2
2
2
2
2
```

Image_pro
File
Home
Share
View
C:\Python27\python.exe
trai...
Search Image_...
R(3).py 21-04-2018 13:11 Python File 2 KB
t(2).py 21-04-2018 13:01 Python File 1 KB
trainner.py 31-03-2018 12:19 Python File 1 KB
Untitled.ipynb 09-04-2018 18:43 IPYNB File 13 KB
Untitled.png 20-04-2018 13:48 PNG File 490 KB
Untitled1.ipynb 10-04-2018 00:40 IPYNB File 5 KB
28 items 1 item selected 686 bytes
ENG
11:49
02-05-2018

3] Detecting in Real Time



Conclusion

The ability to recognize faces is very important to many aspects of life. It not only helps us to recognize those close to us but also allows us to identify individuals we do not know so that we can be more aware of possible dangers. The image of the face captured by web-cam with the help of Processing, OpenCV undergoes different steps as mentioned above. With the normalized test image, the image matching can be simply accomplished by loading a correspondent file from the database, then performing correlation of the test image. The test results show that the detection method used can accurately detect and trace human face in real time.

References

- https://docs.opencv.org/3.0beta/doc/py_tutorials/py_imgproc/py_table_of_contents_imgproc/py_table_of_contents_imgproc.html
- https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_gui/py_video_display/py_video_display.html
- https://www.youtube.com/watch?v=Z78zbnLlPUA&list=PLQVvvaa0QuDdttJXlLtAJxJetJcqmqlQq
- https://www.superdatascience.com/opencv-face-detection/
- https://www.youtube.com/watch?