FaceFilter APP

MITIN SHARMA

INTRODUCTION:

This is a simple application, wherein, webcam capture live image and detect face of the person and put filters like snapchat application.

The main objective of this application is to learn deep learning concepts.

HOW IT WORKS?

This is a python based application

To implement face detection, python application uses openCV library.

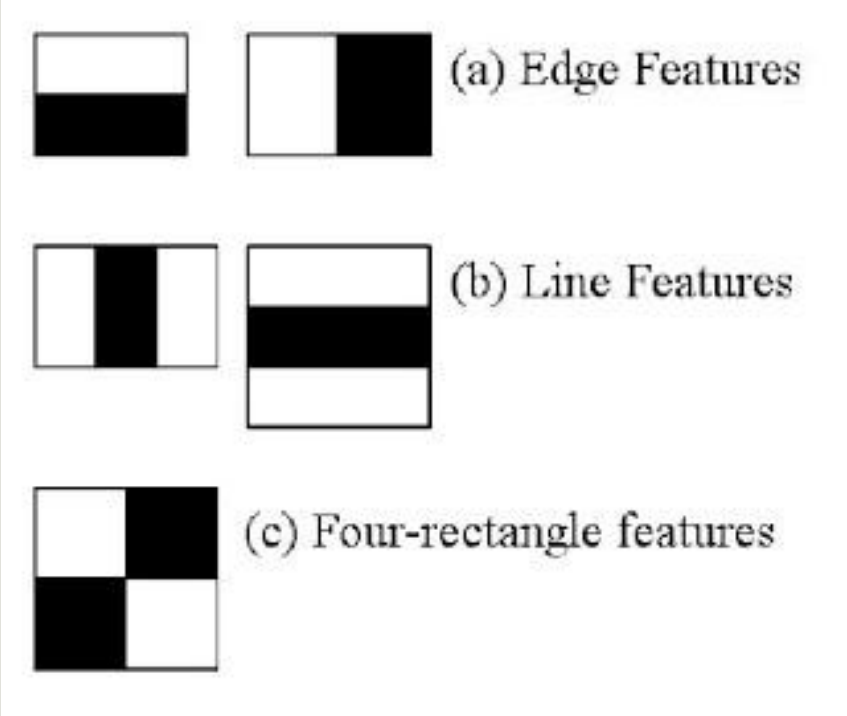
Haar feature based cascade classifiers to learn machine about human faces

Algorithm:

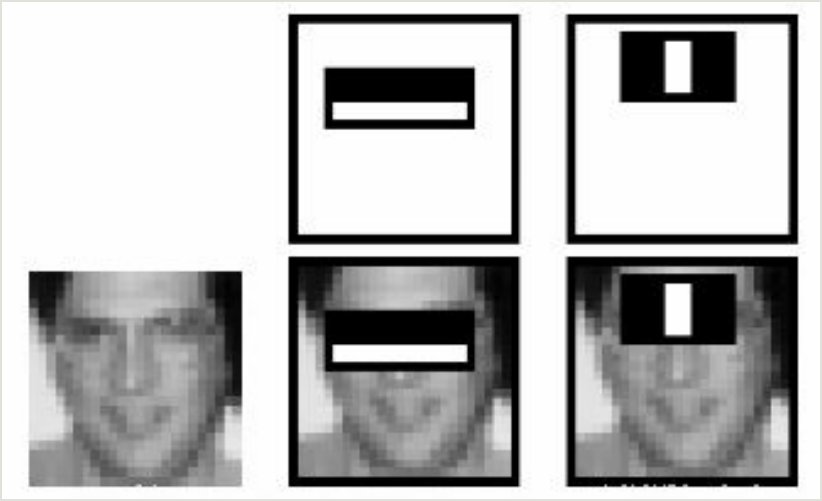
The algorithm needs a lot of positive images (images of faces) and negative images (images without faces) to train the classifier.

Then we need to extract features from it

They are just like our convolutional kernel

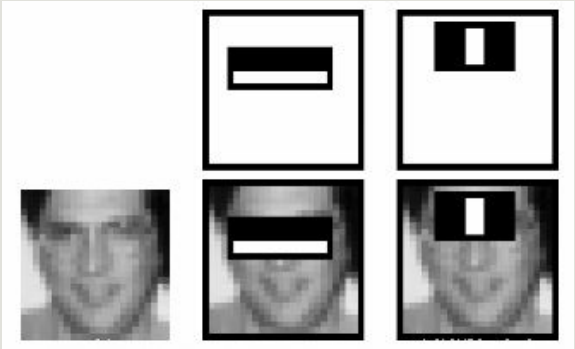


Each feature is a single value obtained by subtracting sum of pixels under white rectangle from sum of pixels under black rectangle



To simplify calculations of sum of pixels we use integral images.

how large may be the number of pixels, to an operation involving just four pixels



Top row shows two good features.

The first feature selected seems to focus on the property that the region of the eyes is often darker than the region of the nose and cheeks.

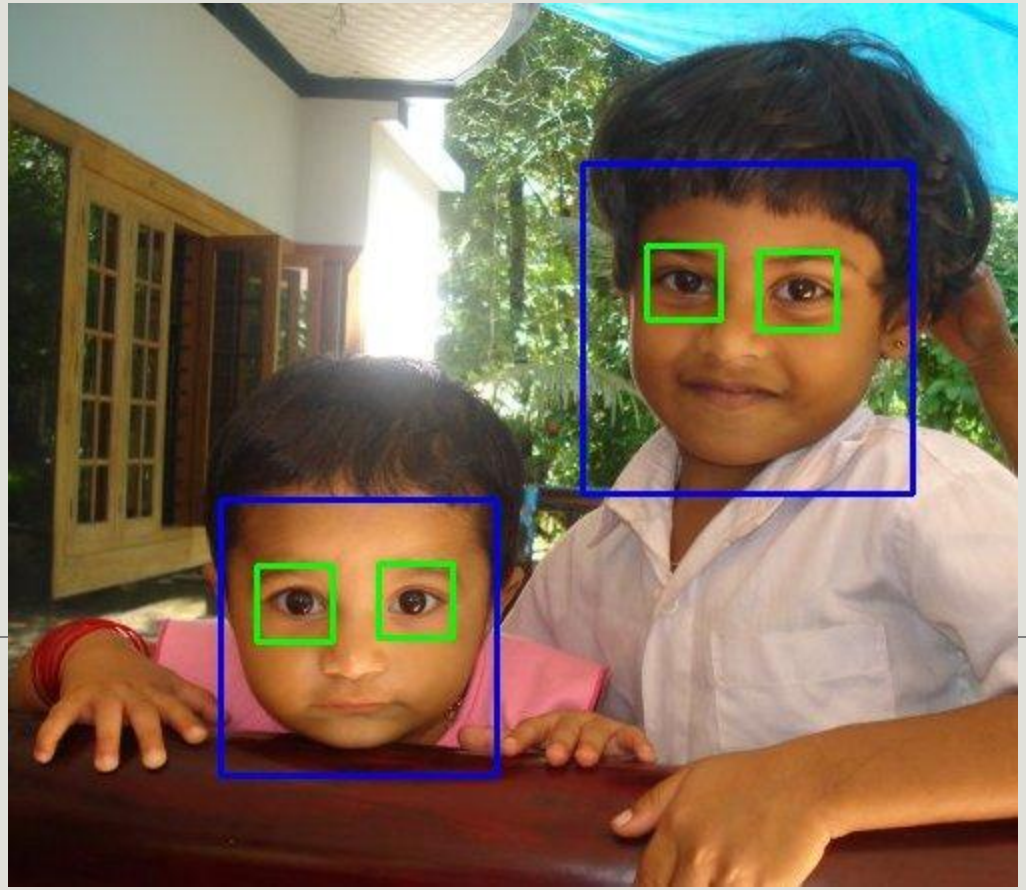
The second feature selected relies on the property that the eyes are darker than the bridge of the nose.

OpenCV:

OpenCV comes with a trainer as well as detector. If you want to train your own classifier for any object like car, planes etc. you can use OpenCV to create one.

OpenCV already contains many pre-trained classifiers for face, eyes, smile etc.

Those XML files are stored in opencv/data/haarcascades/ folder.



Code:

filters.py

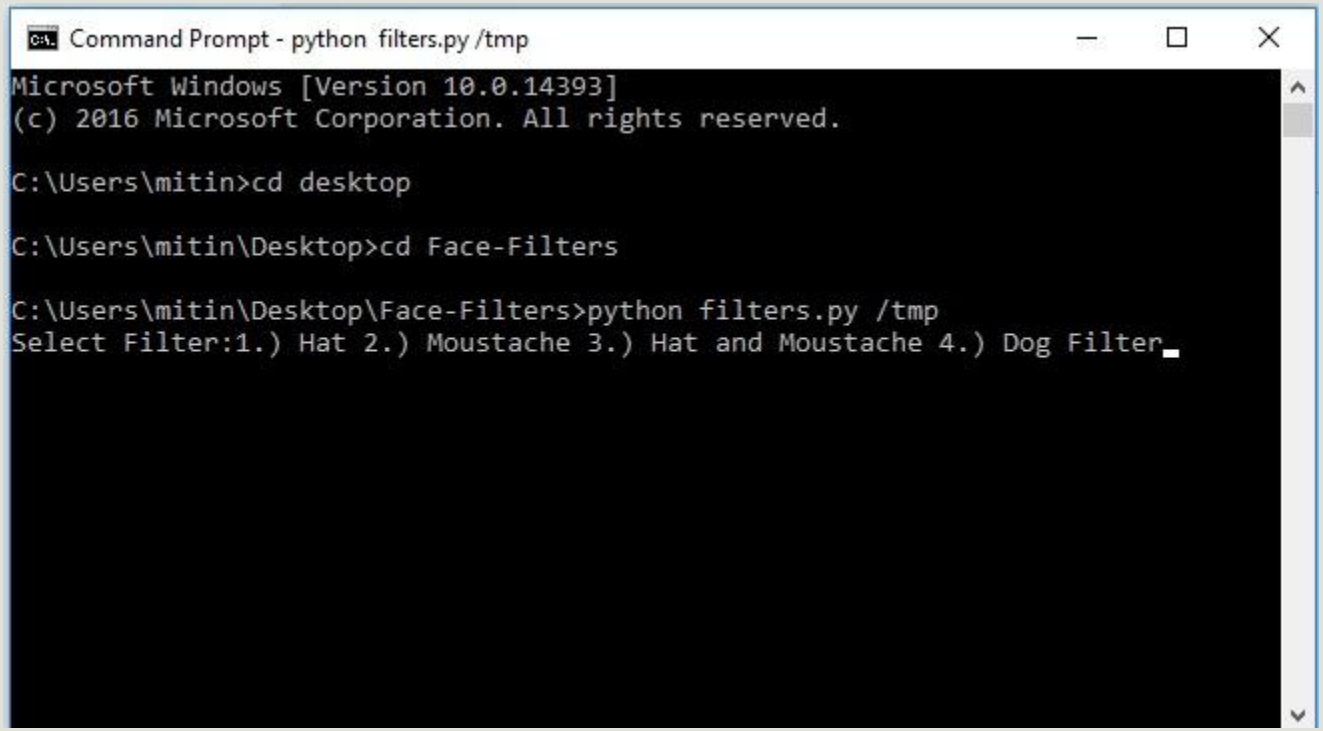
```
# Face filters (Snapchat like) using OpenCV
# @author:- Mitin Sharma

import cv2
import sys
import logging as log
import datetime as dt
from time import sleep
import numpy as np

cascPath = "haarcascade_frontalface_default.xml"  # for face detection
faceCascade = cv2.CascadeClassifier(cascPath)
log.basicConfig(filename='webcam.log',level=log.INFO)

video_capture = cv2.VideoCapture(0)
anterior = 0
mst = cv2.imread('moustache.png')
hat = cv2.imread('cowboy_hat.png')
dog = cv2.imread('dog_filter.png')


def put_moustache(mst,fc,x,y,w,h):

    face_width = w
    face_height = h

    mst_width = int(face_width*0.4166666)+1
    mst_height = int(face_height*0.142857)+1



    mst = cv2.resize(mst,(mst_width,mst_height))
```

filters.py

```
27      mst_width = int(face_width*0.4166666)+1
28      mst_height = int(face_height*0.142857)+1
29
30
31
32      mst = cv2.resize(mst,(mst_width,mst_height))
33
34      for i in range(int(0.62857142857*face_height),int(0.62857142857*face_height)+mst_height):
35          for j in range(int(0.29166666666*face_width),int(0.29166666666*face_width)+mst_width):
36              for k in range(3):
37                  if mst[i-int(0.62857142857*face_height)][j-int(0.29166666666*face_width)][k] <235:
38                      fc[y+i][x+j][k] = mst[i-int(0.62857142857*face_height)][j-int(0.29166666666*face_width)][k]
39      return fc
40
41  def put_hat(hat,fc,x,y,w,h):
42
43      face_width = w
44      face_height = h
45
46      hat_width = face_width+1
47      hat_height = int(0.35*face_height)+1
48
49      hat = cv2.resize(hat,(hat_width,hat_height))
50
51      for i in range(hat_height):
52          for j in range(hat_width):
53              for k in range(3):
54                  if hat[i][j][k]<235:
55                      fc[y+i-int(0.25*face_height)][x+j][k] = hat[i][j][k]
56      return fc
57
58  def put_dog_filter(dog,fc,x,y,w,h):
```

SCREENS:

HOME SCREEN:-


Command Prompt - python filters.py /tmp
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.
C:\Users\mitin>cd desktop
C:\Users\mitin\Desktop>cd Face-Filters
C:\Users\mitin\Desktop\Face-Filters>python filters.py /tmp
Select Filter:1.) Hat 2.) Moustache 3.) Hat and Moustache 4.) Dog Filter

Hat Filter:-

Moustache Filter:-

Hat and Moustache Filter:-

Dog Filter:-

THANK YOU