CS512 - Computer Vision

Prof. Dr. Gady Agam

Project Report

Smile and Frown Detection

Adithya Sreenath Chandrashekarapuram
A20402135

Alisha Anjum Aleem
A20379905

Overview

The difference between a `bad' photo and a `good' photo is often a matter of whether or not the person in the photo is smiling or frowning. With the help of feature recognition, smiles and frown can be identified in a photo. We want to automatically detect a smiling subject in a picture. Our intended use is in the digital photography industry, where this program can be applied to automatically select the best frame in a set of similar frames.

Keywords: Haar-Cascade**,** feature detection, feature recognition, face detection, eye-brow detection, smile detection, frown detection.

Introduction

The most common facial expressions which we observe in humans is a smile and a frown. The smile gives a favorable expression on others and makes one more approachable whereas a frown does just the opposite. A smile bespeaks a person joy, felicity, admiration or gratification. However a frown depicts a person's misery, grief or sadness. The mental state of a person is judged by detecting smiles or frowns. It has many applications in video cameras, mobile phones, security systems, distance learning systems, video conferencing, interactive systems like gaming etc. The detection starts with the facial recognition. The main advantage of the smile detection is it ensures whether smiles are detected by the camera or not. Generally in the present system, capturing the decision boundary of spontaneous expressions is very difficult.

Implementation

There are 8 parts for the design, i.e., pre-processing and expression detection, respectively. The preprocessing includes:

1) Training the Haar-classifier
    - First we take a dataset of images and place them in directories.
    - We then classify them into positive and negative images.
    - We create an info file and use the opencv_createsamples function to create vector files for positive images.
    - We create a bg.txt file from the negative samples.
    - Finally we train the classifier using the opencv_traincascade function.
    - The result is a haar_cascade.xml file used for detection of features.

2) Capturing the image

- First we use cv2.VideoCapture() function to start a video capture.
- We then retrieve a single frame by using the ret, frame = cap.read() function.

3) Face detection

- We take the frame and convert it into gray scale by using the cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY) statement.
- We then pass this image into a haar classifier function which compares the image with a database of positive and negative images. We use the following function to compare:

```
Face = faceCascade.detectMultiScale(gray,scaleFactor=sF,
        minNeighbors=8,minSize=(55, 55),flags=0)
```

- Upon comparison we can determine whether the image contains a face or not.

4) Smile Detection

- Upon detecting the face we isolate this region and perform smile detection.
- First we take the isolated face region and pass the image to a haar classifier function which checks with a list of positive and negative images. We use this function to check for a smile:

```
Smile = smileCascade.detectMultiScale(roi_gray,scaleFactor=1.7,
          minNeighbors=22,minSize=(25, 25),flags=0)
```

- After comparison we set a threshold for the smile and display a rectangle around the mouth region.

```
if len(smile) > 0.75:
        for (x, y, w, h) in smile:
           cv2.rectangle(roi_color, (x, y), (x+w, y+h), (255, 0, 0), 2)
```

5) Eyebrow Detection

- Upon detecting the face we isolate this region and perform eyebrow detection.
- First we take the isolated face region and pass the image to a haar classifier function which checks with a list of positive and negative images of eyes. We use this function to check for eyes:

```
Eyes = eye_cascade.detectMultiScale(roi_gray)
```

- Upon determining the position of the eyes we can detect the eyebrows and draw a rectangle around them.

6) Frown Detection

- After we detect the eyebrows we check for the position of the eyebrows.
- We then check for the presence of the smile.
- After setting a threshold we can determine whether a person is frowning or not based on the 2 factors.
- We then display a rectangle around the eyebrow region to determine that the person is frowning.

7) Display text over Detected Region

- We use the following code to display text over the rectangle upon detection of a smile.

```
cv2.putText(roi_color,"Smile  Found",(x,y+h+20),font,0.50,(255,0,0),2)
```

- We follow the same procedure for the remaining detections.

8) Emotion Detection

- After we determine whether a person is smiling or frowning we can check if he/she is happy or sad.
- If the person is smiling he is said to be happy.
- If the person is frowning he is said to be sad.

Process Diagram

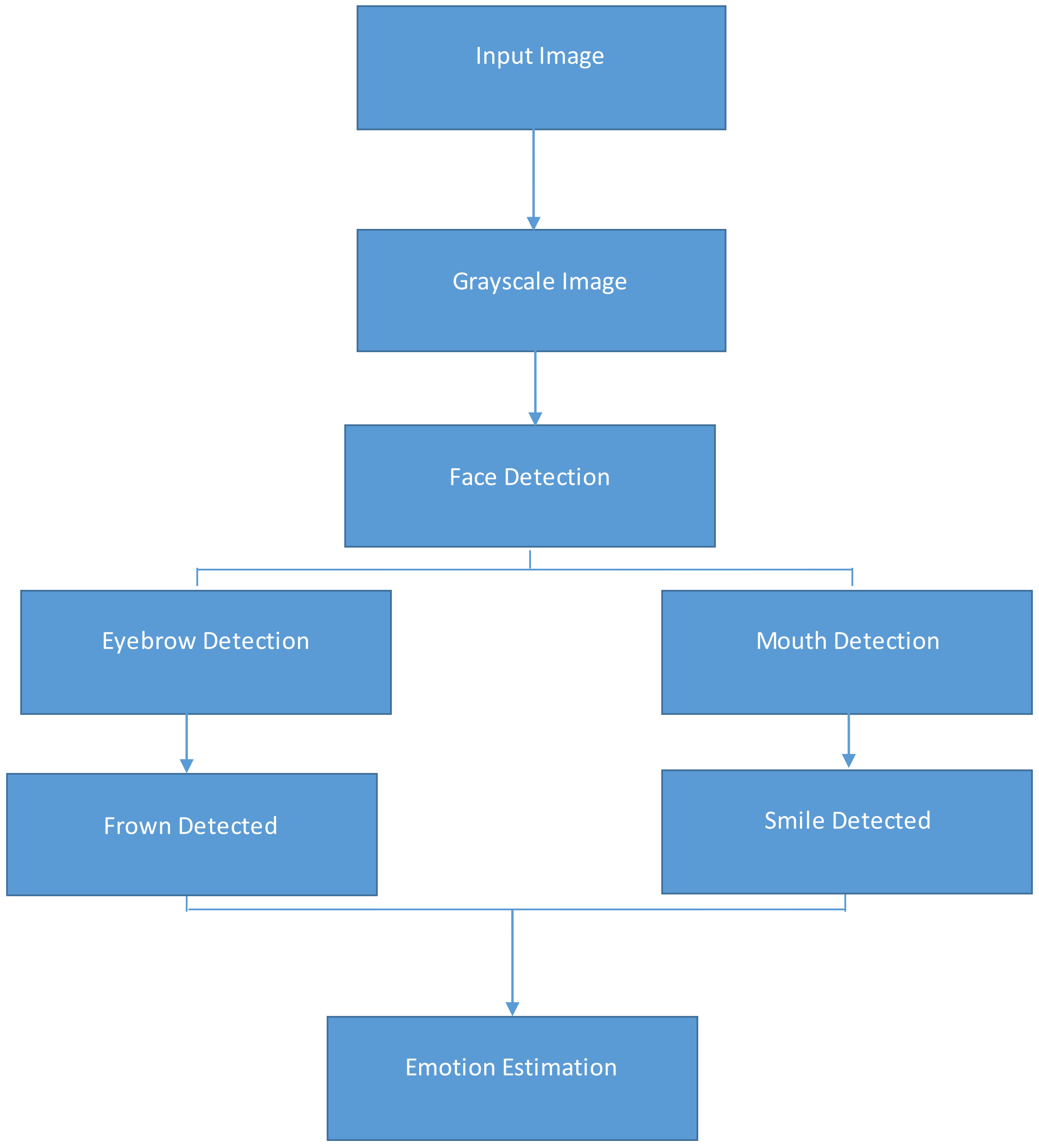
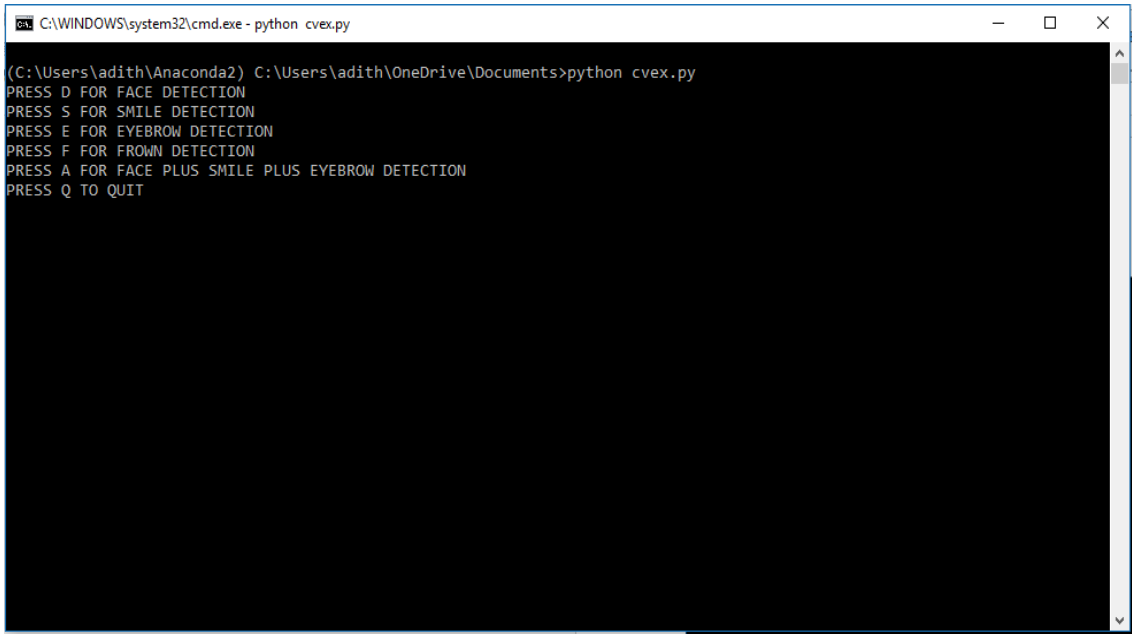


Fig.1 Smile and Frown Detection

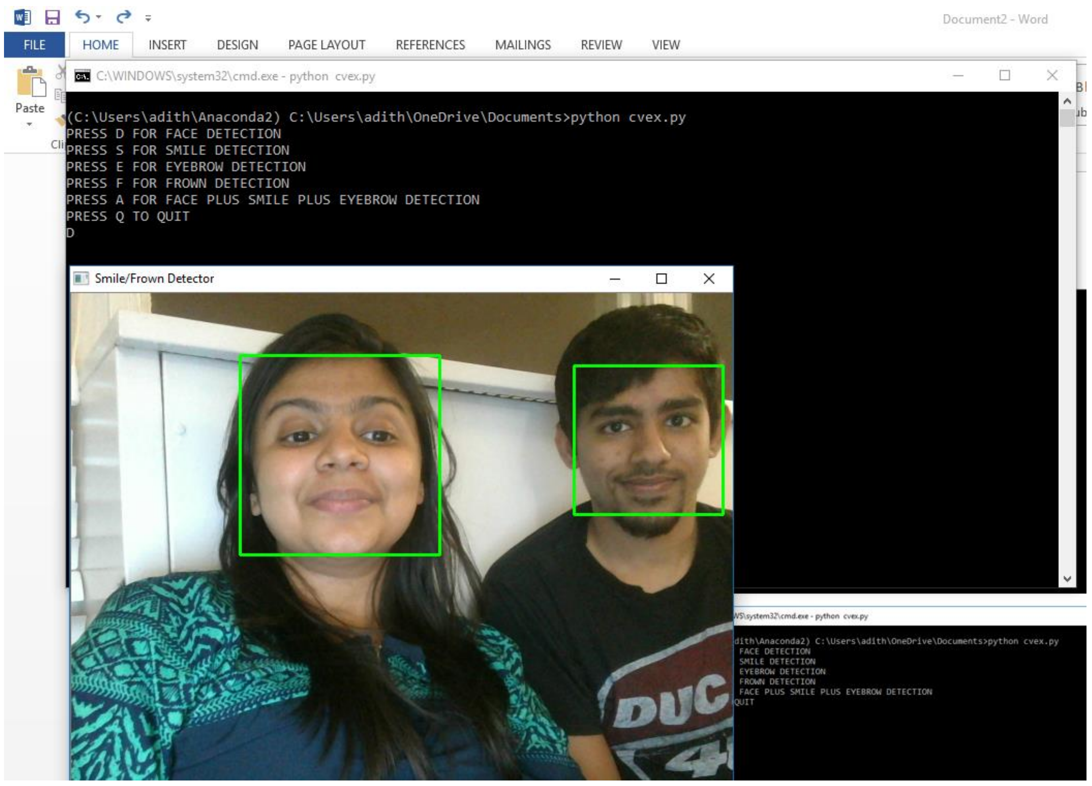
Results

SELECTION OF INPUT FOR DETECTION:



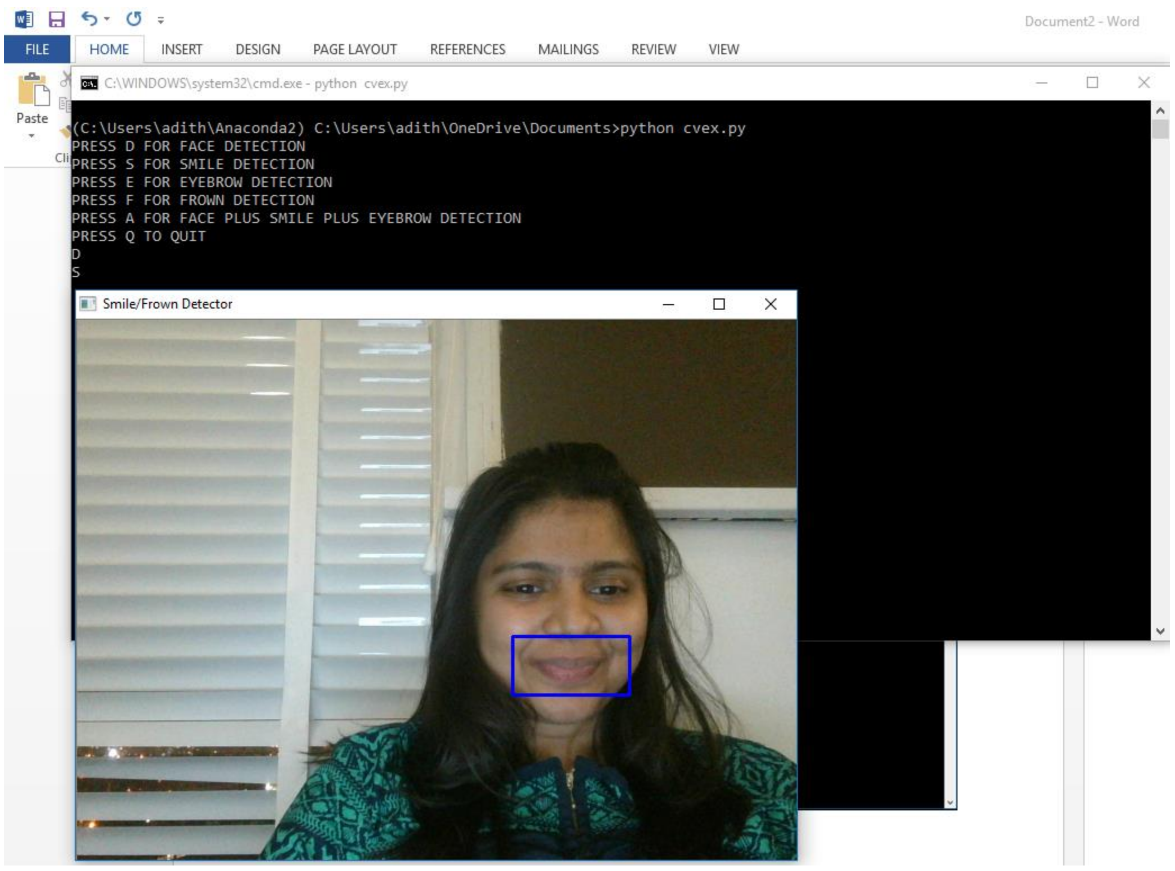
```
(C:\Users\adith\Anaconda2) C:\Users\adith\OneDrive\Documents>python cvex.py
PRESS D FOR FACE DETECTION
PRESS S FOR SMILE DETECTION
PRESS E FOR EYEBROW DETECTION
PRESS F FOR FROWN DETECTION
PRESS A FOR FACE PLUS SMILE PLUS EYEBROW DETECTION
PRESS Q TO QUIT
```

FACE DETECTION:



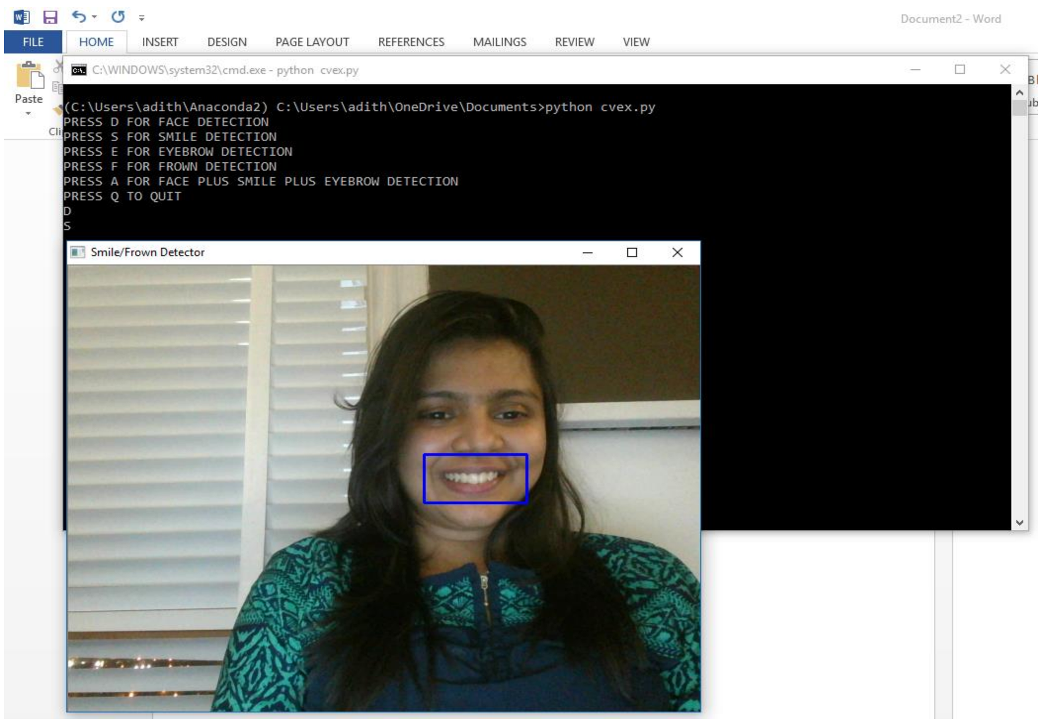
```
(C:\Users\adith\Anaconda2) C:\Users\adith\OneDrive\Documents>python cvex.py
PRESS D FOR FACE DETECTION
PRESS S FOR SMILE DETECTION
PRESS E FOR EYEBROW DETECTION
PRESS F FOR FROWN DETECTION
PRESS A FOR FACE PLUS SMILE PLUS EYEBROW DETECTION
PRESS Q TO QUIT
D
```

SMILE DETECTION:



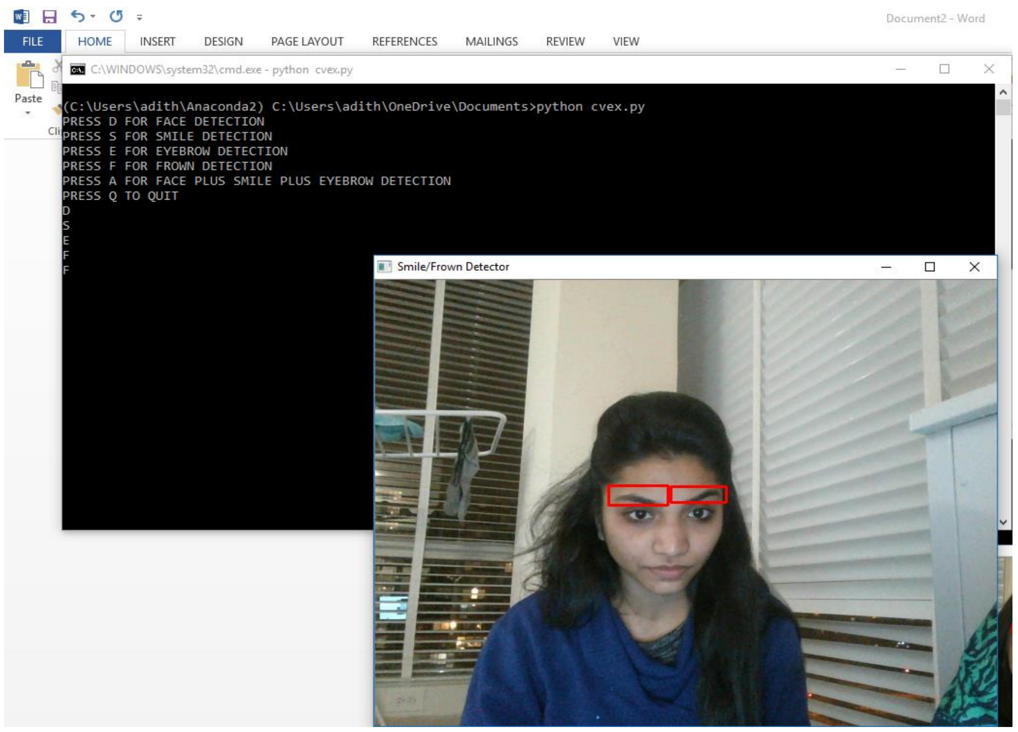
```
(C:\Users\adith\Anaconda2) C:\Users\adith\OneDrive\Documents>python cvex.py
PRESS D FOR FACE DETECTION
PRESS S FOR SMILE DETECTION
PRESS E FOR EYEBROW DETECTION
PRESS F FOR FROWN DETECTION
PRESS A FOR FACE PLUS SMILE PLUS EYEBROW DETECTION
PRESS Q TO QUIT
D
S
```

SMILE DETECTION:



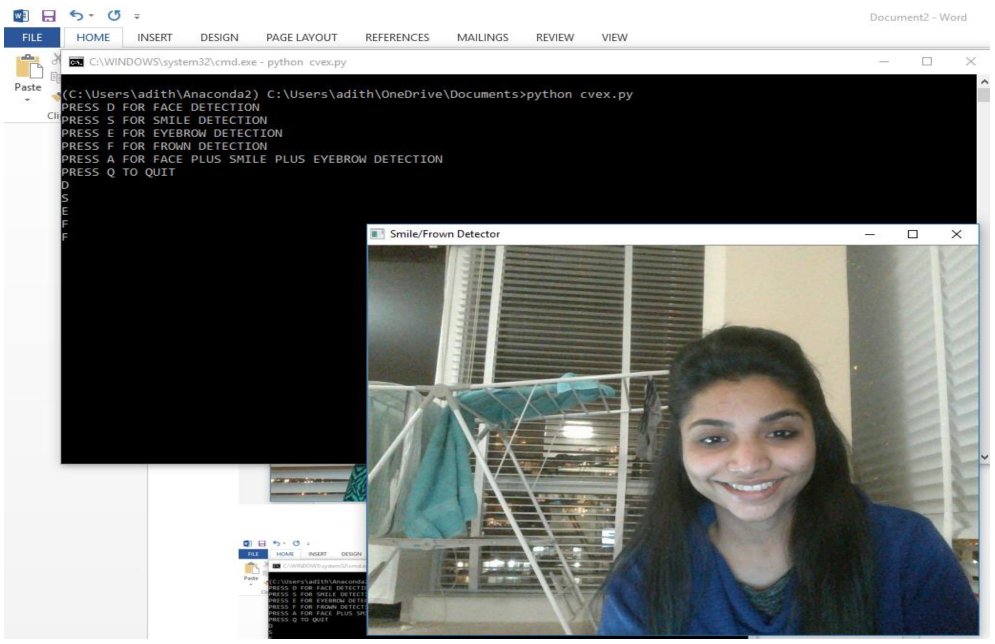
```
(C:\Users\adith\Anaconda2) C:\Users\adith\OneDrive\Documents>python cvex.py
PRESS D FOR FACE DETECTION
PRESS S FOR SMILE DETECTION
PRESS E FOR EYEBROW DETECTION
PRESS F FOR FROWN DETECTION
PRESS A FOR FACE PLUS SMILE PLUS EYEBROW DETECTION
PRESS Q TO QUIT
D
S
```

FROWN DETECTION:



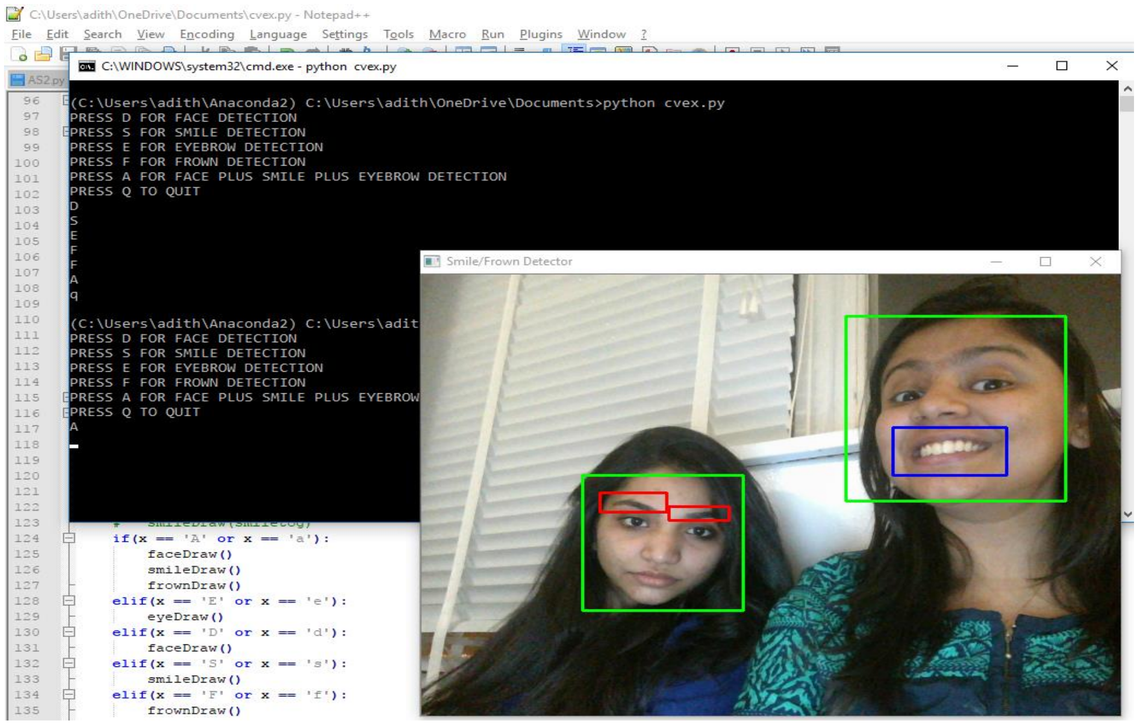
```
(C:\Users\adith\Anaconda2) C:\Users\adith\OneDrive\Documents>python cvex.py
PRESS D FOR FACE DETECTION
PRESS S FOR SMILE DETECTION
PRESS E FOR EYEBROW DETECTION
PRESS F FOR FROWN DETECTION
PRESS A FOR FACE PLUS SMILE PLUS EYEBROW DETECTION
PRESS Q TO QUIT
D
S
E
F
F
```

NO EYEBROW DETECTION DUE TO PRESENCE OF SMILE:



```
(C:\Users\adith\Anaconda2) C:\Users\adith\OneDrive\Documents>python cvex.py
PRESS D FOR FACE DETECTION
PRESS S FOR SMILE DETECTION
PRESS E FOR EYEBROW DETECTION
PRESS F FOR FROWN DETECTION
PRESS A FOR FACE PLUS SMILE PLUS EYEBROW DETECTION
PRESS Q TO QUIT
D
S
E
F
F
```

DETECTING SMILE AND FROWN:


C:\Users\adith\OneDrive\Documents\cvex.py - Notepad++
File
Edit
Search
View
Encoding
Language
Settings
Tools
Macro
Run
Plugins
Window
?
AS2.py
C:\WINDOWS\system32\cmd.exe - python cvex.py
Smile/Frown Detector

```
(C:\Users\adith\Anaconda2) C:\Users\adith\OneDrive\Documents>python cvex.py
PRESS D FOR FACE DETECTION
PRESS S FOR SMILE DETECTION
PRESS E FOR EYEBROW DETECTION
PRESS F FOR FROWN DETECTION
PRESS A FOR FACE PLUS SMILE PLUS EYEBROW DETECTION
PRESS Q TO QUIT
D
S
E
F
F
A
q

(C:\Users\adith\Anaconda2) C:\Users\adit
PRESS D FOR FACE DETECTION
PRESS S FOR SMILE DETECTION
PRESS E FOR EYEBROW DETECTION
PRESS F FOR FROWN DETECTION
PRESS A FOR FACE PLUS SMILE PLUS EYEBROW
PRESS Q TO QUIT
A
```

```python
#    smileDraw(smilecog)
    if(x == 'A' or x == 'a'):
        faceDraw()
        smileDraw()
        frownDraw()
    elif(x == 'E' or x == 'e'):
        eyeDraw()
    elif(x == 'D' or x == 'd'):
        faceDraw()
    elif(x == 'S' or x == 's'):
        smileDraw()
    elif(x == 'F' or x == 'f'):
        frownDraw()
```

Future Enhancements

The program designed can be improved in the future in the following ways:

1) A better haar-cascade can be developed by using a large database of images to improve the efficiency of detection.
2) A separate classifier can be developed to train images for any emotion. For example: Anger, Surprise, Winking, etc can be detected.
3) GUI can be implemented

References

[1] An-Chao Tsai1, Ting-WeiLin,Ta-Wen Kuan,K.Bharanitharan, Jih-Tso Chang and Jhing-Fa Wang, An Efficient Smile and Frown Detection Algorithm', International Conference on Orange Technologies (ICOT),2015

[2] https://docs.opencv.org/3.3.0/d7/d8b/tutorial_py_face_detection.html