

# Table of Contents

Abstract: .....	2
Introduction:.....	2
Motivation:.....	2
Features: .....	3
Components: .....	3
Hardware Components:.....	3
Software Components: .....	3
Work back-down among team member:.....	4
Working process: .....	4
Flow Chart diagram: .....	4
Code: .....	5
Result: .....	5
Future scope: .....	5
Implementation image: .....	6
Conclusion: .....	6
References:.....	7

## **Abstract:**

Identifying a person face from an image has been popularized through the mass media. This report describes the face detection project. It reports the technologies available in the Open-Computer-Vision (OpenCV) library and methodology to implement those using Python. For face detection, Haar-Cascades were used. Next, the results are shown including screen-shots which shows detected faces from an image. The reports concluded with the authors' opinion on the project and possible applications.

## **Introduction:**

The following document is a report on the mini project for Robotic visual perception and autonomy. It involved building a system for face detection using several classifiers available in the open computer vision library (OpenCV). Face detection is used to identify a face from an image. This is followed by the explanation of HAAR-cascades. Next, the methodology and the results of the project are described. A discussion regarding the challenges and the resolutions are described. Finally, conclusion is provided on the pros and cons of each algorithm and possible implementations.

## **Motivation:**

- To identify a human face from an image is the main purpose
- To study OpenCV and implement it in this project.
- To understand how face detection method works.

## Features:

- It can detect all the faces from an image.
- It is easy to use.
- It can work on low configured computer
- It does not use too much disk space
- Its requirement is easy to get and all software is free in online.

## Components:

### Hardware Components:

- A computer
- Processor minimum Intel Pentium 4
- 512mb ram
- Hard disk storage 32gb (minimum)

### Software Components:

- Operating system : Windows 7
- Python 2.7.6
- Numpy 1.10.2
- Matplotlib-1.5.0
- OpenCV 2.4

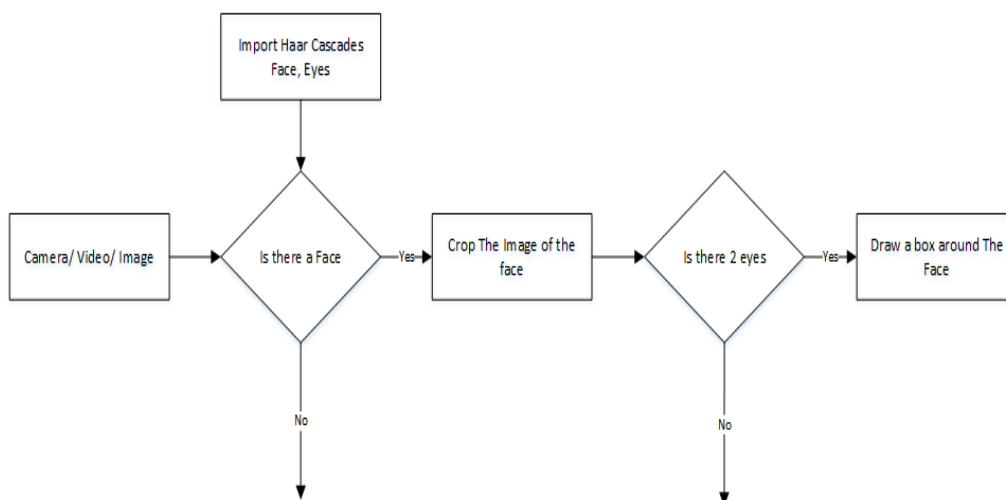
## Work back-down among team member:

Name:	ID:	Work:
1)MD Shawon Hossain	153402351	Implementation ,testing and project report
2)MD Faisal Ahmed	152392326	Coding, testing and project report
3)MD Abu Bakkar	153402348	Testing and project report

## Working process:

- First we write the code in a file and save it as FaceDetect.py.
- Then we launch python IDLE from our computer.
- We already wrote the image name on the FaceDetect.py file.
- After running the module, we can see a rectangle around the face.
- This indicates there is a human face on the image.

## Flow Chart diagram:



## Code:

```
import cv2

import numpy as np

img = cv2.imread('1.jpg')

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

gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)

faces = face_csc.detectMultiScale(gray,1.1,4)

for(x,y,w,h) in faces:

    cv2.rectangle(img,(x,y),(x+w,y+h),(0,255,0),3)

cv2.imshow('img',img)

cv2.waitKey(0)
```

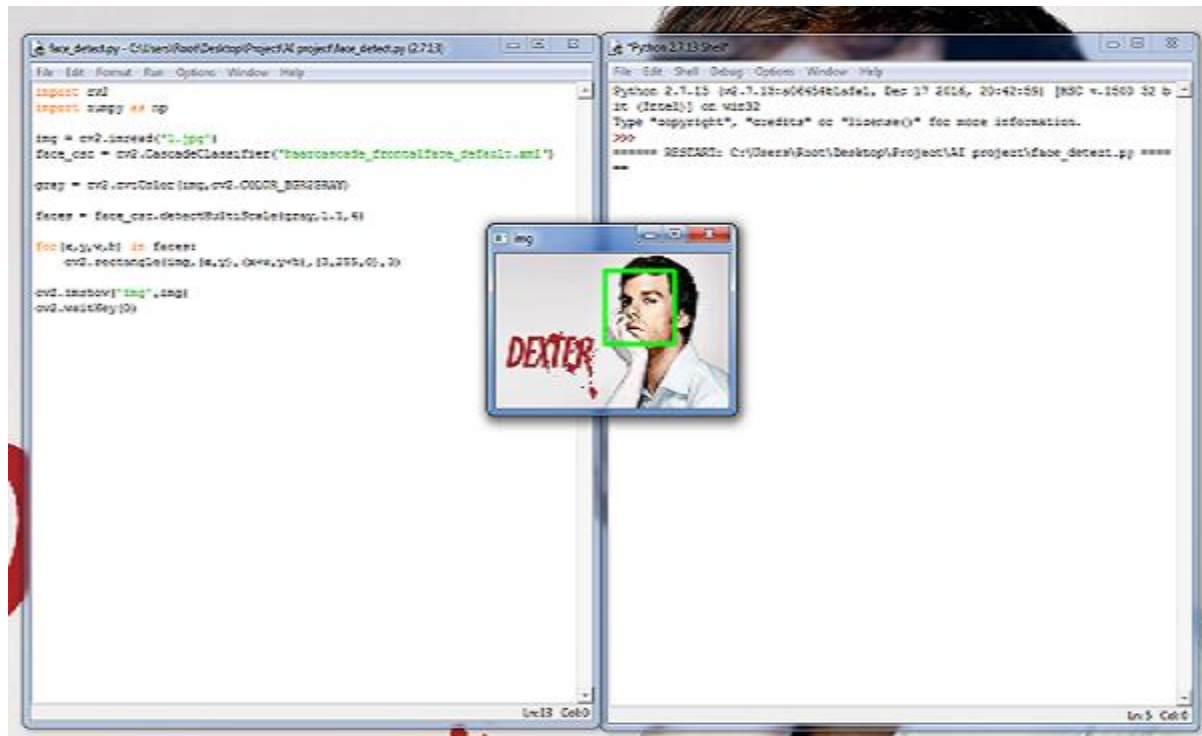
## Result:

Our project was able to detect the faces from any images successfully.

## Future scope:

- We will add image recognition to our next update.
- We will make a device with camera which will able to detect and recognize the person.

## Implementation image:



## Conclusion:

This paper describes the project for visual perception and autonomy module. Next, it explains the technologies used in the project and the methodology used. Finally, it shows the results, discuss the challenges and how they were resolved followed by a discussion. Using Haar-cascades for face detection worked extremely well even when subjects wore spectacles. Considering all factors combined with Haar-cascades can be implemented as a cost effective face detection platform.

## References:

- [1] Takeo Kanade. Computer recognition of human faces, volume 47. Birkhäuser Basel, 1977.
- [2] Lawrence Sirovich and Michael Kirby. Low-dimensional procedure for the characterization of human faces. *Josa a*, 4(3):519{524, 1987.
- [3] X. Wang, T. X. Han, and S. Yan. An hog-lbp human detector with partial occlusion handling. In 2009 IEEE 12th International Conference on Computer Vision, pages 32{39, Sept 2009.
- [4] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceed-ings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. CVPR 2001, volume 1, pages I{511{I{518 vol.1, 2001.
- [5] Rainer Lienhart and Jochen Maydt. An extended set of haar-like features for rapid object detection. In *Image Processing. 2002. Proceedings. 2002 International Conference on*, volume 1, pages I{I. IEEE, 2002.
- [6] John P Lewis. Fast template matching. In *Vision interface*, volume 95, pages 15{19, 1995.