

**Project name: Image processing in face and eyes detection used open cv**

**Submitted To:**

**Supta Richard Philip**

**Senior Lecturer (CSE Department)**

**City University, Dhaka.**

**Submitted By:**

|  |  |
| --- | --- |
| **Name:** | **ID:** |
| **1.Rasel Ahmed Sifat** | **151382356** |
| **2.Md Riad Mia** | **151382337** |
| **3.** **Saher Banu** | 143372002 |
| **4.** **Mehedi hasan** | **163432535** |

**Submission Date:13/3/2019**

**Table of Contents**

[Abstract: 3](#_Toc3544248)

[Introduction: 3](#_Toc3544249)

[Motivation: 3](#_Toc3544250)

[Features: 4](#_Toc3544251)

[Components: 4](#_Toc3544252)

[Hardware Components: 4](#_Toc3544253)

[Software Components: 4](#_Toc3544254)

[Work back-down among team member: 5](#_Toc3544255)

[Working process: 5](#_Toc3544256)

[Flow Chart diagram: 5](#_Toc3544257)

[Code: 6](#_Toc3544258)

[Result: 6](#_Toc3544259)

[Future scope: 7](#_Toc3544260)

[Implementation image: 7](#_Toc3544261)

[Conclusion: 7](#_Toc3544262)

[References: 8](#_Toc3544263)

# Abstract:

Identifying a person face and eyes from an image has been popularized through the mass media. This report describes the face and eyes detection project. It reports the technologies available in the OpenComputer-Vision (OpenCV) library and methodology to implement those using Python. For face and eyes detection, Haar-Cascades were used. Next, the results are shown including screen-shots which shows detected face and eyes and eyes 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 and eyes detection using several classifiers available in the open computer vision library (OpenCV). Face and eyes detection is used to identify a face and eyes 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 and eyes from an image is the main purpose
* To study OpenCV and implement it in this project.
* To understand how face and eyes detection method works.

# Features:

* It can detect all the face and eyes 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)

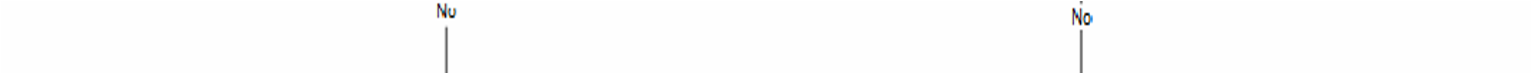
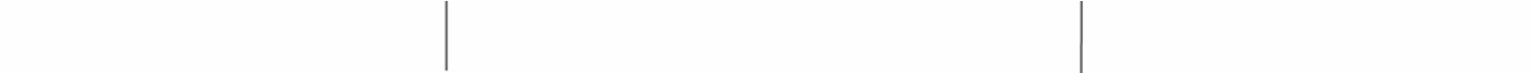
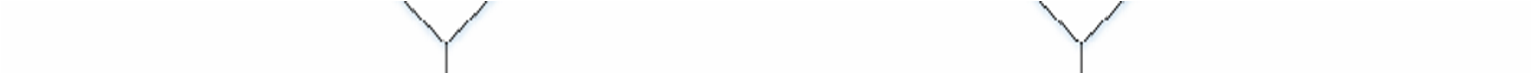
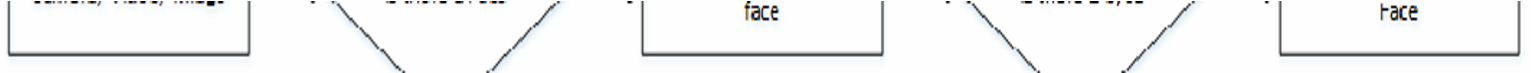
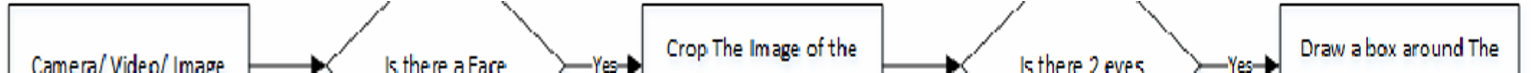
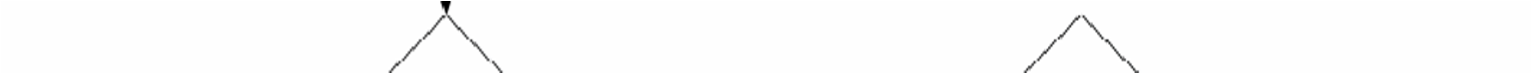
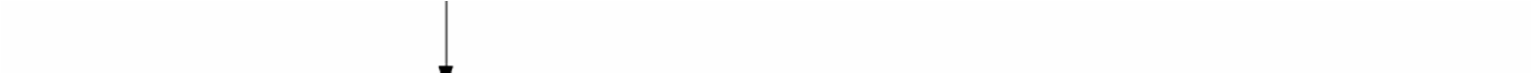
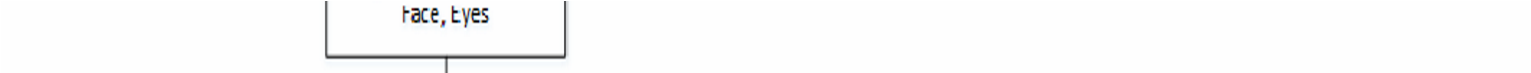
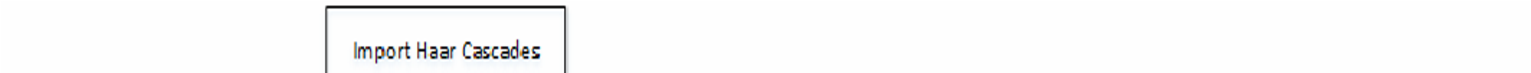
## Software Components:

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

# Working process:

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

# Flow Chart diagram:



# Code:

import cv2

face\_cascade = cv2.CascadeClassifier('haarcascade\_frontalface\_default.xml')

eye\_cascade = cv2.CascadeClassifier('haarcascade\_eye.xml')

cap = cv2.VideoCapture(0)

while 1:

ret, img = cap.read() gray = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)

faces = face\_cascade.detectMultiScale(gray, 1.3, 5)

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

cv2.rectangle(img,(x,y),(x+w,y+h),(255,255,0),2)

roi\_gray = gray[y:y+h, x:x+w]

roi\_color = img[y:y+h, x:x+w]

eyes = eye\_cascade.detectMultiScale(roi\_gray )

for (ex,ey,ew,eh) in eyes:

cv2.rectangle(roi\_color,(ex,ey),(ex+ew,ey+eh),(0,127,255),2)

cv2.imshow('img',img)

k = cv2.waitKey(30) & 0xff

if k == 27:

break

cap.release()

cv2.destroyAllWindows()

# Result:

Our project was able to detect the face and eyess 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.

# 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 and eyes detection worked extremely well even when subjects wore spectacles. Considering all factors combined with Haar-cascades can be implemented as a cost effective face and eyes detection platform.

# References:

1. Takeo Kanade. Computer recognition of human face and eyess, volume 47. Birkh auser Basel, 1977.
2. Lawrence Sirovich and Michael Kirby. Low-dimensional procedure for the characterization of human face and eyess. 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.

1. 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.

1. 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.

1. John P Lewis. Fast template matching. In Vision interface and eyes, volume 95, pages 15{19, 1995.