**Face and Eye detection using OpenCv With Raspberry pi:**

**Project Description:**

In this project we are using OpenCv in Raspberry pi. This project is used to detect the human Face and eye with the help of OpenCv tool. In order to do object detection with cascade files, you first need cascade files. For the extremely popular tasks, these file already exist.

**Software Required:**

* OpenCv
* Raspian OS

**Hardware Used:**

* Raspberry Pi
* Camera
* Power Supply

**OpenCv:**

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code. The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects and extract 3D models of objects.

**Installation steps for python-OpenCv**:

sudo apt-get update

sudo apt-get upgrade

sudo apt-get install build-essential

sudo apt-get install cmake git libgtk2.0-dev pkg-config libavcodec-dev libavformat-dev libswscale-dev

sudo apt-get install python-dev python-numpy libtbb2 libtbb-dev libjpeg-dev libpng-dev libtiff-dev libjasper-dev libdc1394-22-dev

sudo apt-get install python-opencv

sudo apt-get install python-matplotlib

**Circuit Diagram:**



**Python Code:**

import io

import picamera

import cv2

import numpy

#Create a memory stream so photos doesn't need to be saved in a file

stream = io.BytesIO()

#Get the picture (low resolution, so it should be quite fast)

#Here you can also specify other parameters (e.g.:rotate the image)

with picamera.PiCamera() as camera:

camera.resolution = (320, 240)

camera.capture(stream, format='jpeg')

#Convert the picture into a numpy array

buff = numpy.fromstring(stream.getvalue(), dtype=numpy.uint8)

#Now creates an OpenCV image

img = cv2.imdecode(buff, 1)

#Load a cascade file for detecting faces

face\_cascade = cv2.CascadeClassifier('/home/pi/Desktop/opencv/faces.xml')

eye\_cascade = cv2.CascadeClassifier('/home/pi/Desktop/opencv/haarcascade\_eye.xml')

#Convert to grayscale

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

#Look for faces in the image using the loaded cascade file

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

print "Found "+str(len(faces))+" face(s)"

#Draw a rectangle around every found face

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

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

#Save the result2 image

cv2.imwrite('result2.jpg',img)