

**S R M INSTITUTE OF SCIENCE & TECHNOLOGY**

**(FACULTY OF SCIENCE & HUMANITIES)**

DEPARTMENT OF COMPUTER APPLICATIONS

**Lab Manual**

**REAL WORLD COMPUTER VISION APPLICATION**

**[UDS21502J]**

**BCA 3rd YEAR, 5th SEMESTER**

**Session: 2024-25**

|  |  |  |
| --- | --- | --- |
| **SUBMITTED TO :** | **SUBMITTED BY :** | |
| **Aalekh Choudhary** | | **Student Reg No** | |
| **Assistant Professor** | | **Student Name** | |
| **Department of Computer Applications** | **Department of Computer Applications** | |

S R M INSTITUTE OF SCIENCE & TECHNOLOGY

DELHI NCR CAMPUS, MODINAGAR

**(FACULTY OF SCIENCE & HUMANITIES)**

DEPARTMENT OF COMPUTER APPLICATIONS

**Register No. :**

# **BONAFIDE CERTIFICATE**

Certified to be the Bonafide record of the work done by **XYZ Kumar** of BCA **1st Year/ 2nd** Semesterfor the Award of **Bachelor’s Degree** in the **DEPARTMENT OF COMPUTER APPLICATIONS** in **Real World Computer Vision Application [ UDS21502J ]** laboratory during the Academic Year **2024-25.**

**SUBJECT IN – CHARGE HEAD – COMPUTER APPLICATIONS**

**Submitted for the University Examination held on \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.**

**INTERNAL EXAMINER - 1 INTERNAL EXAMINER - 2**

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **S.**  **No** | **Experiment Title** | **Page No** |
| 1 | Write a code/program in OpenCV(Using Python) to read and display an image. |  |
| 2 | Write a code/program in OpenCV(Using Python) to read , resize and display an image. |  |
| 3 | Write a program in OpenCV(Using Python) to show multiple images in a single frame. |  |
| 4 | Write a program in OpenCV(Using Python) crop an image also show its original image. |  |
| 5 | Write a program in OpenCV using python to detect edges of an image |  |
| 6 | Write a program in OpenCV for detecting a face by using AI code. |  |
| 7 | Write a program in OpenCV for detecting eyes on face by using AI code. |  |
| 8 |  |  |
| 9 |  |  |
| 10 |  |  |
| 11 |  |  |
| 12 |  |  |
| 13 |  |  |
| 14 |  |  |
| 15 |  |  |

**Experiment - 1**

**Experiment 1: Write a code/program in OpenCV(Using Python) to read and display an image.**

***Aim: Using OpenCV library of python , insert an image and display it.***

**Algorithm/ Procedure:**

1. Start Program by opening Jupiter Navigator
2. Open Jupiter Notepad
3. Import CV2 library
4. Insert path of an image to read.
5. Show image
6. Use waitKey( ) function for exit
7. Exit

**Program:**

Import cv2

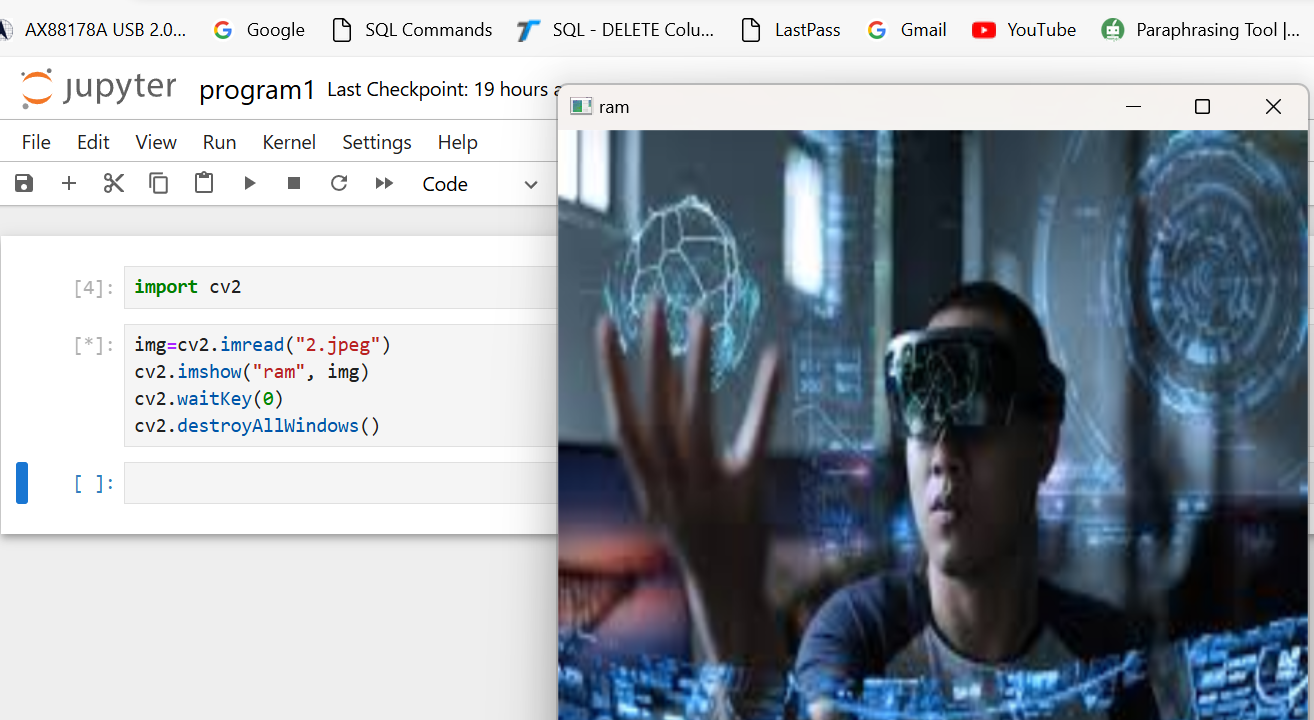
img=cv2.imread("2.jpeg") #reading an image by using a path

cv2.imshow("ram", img) #display image by using imshow function

cv2.waitKey(0) # display a window for given milliseconds or until any key is pressed

cv2.destroyAllWindows() #allows users to destroy or close all windows

**Input/ Output**

****

**Result: program successfully executed, Image successfully read and displayed.**

.

**Experiment 2**

**Experiment 2: Write a code/program in OpenCV(Using Python) to read , resize and display an image.**

***Aim: Using OpenCV library of python , insert an image and display it.***

**Algorithm/ Procedure:**

1. Start Program by opening Jupiter Navigator
2. Open Jupiter Notepad
3. Import CV2 library
4. Insert path of an image to read.
5. Put resize function for resizing image
6. Show image
7. Use waitKey( ) function for exit
8. Exit

**Program:**

import cv2

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

reimg=cv2.resize(img, (500,500)) #used function for resizing its dimension

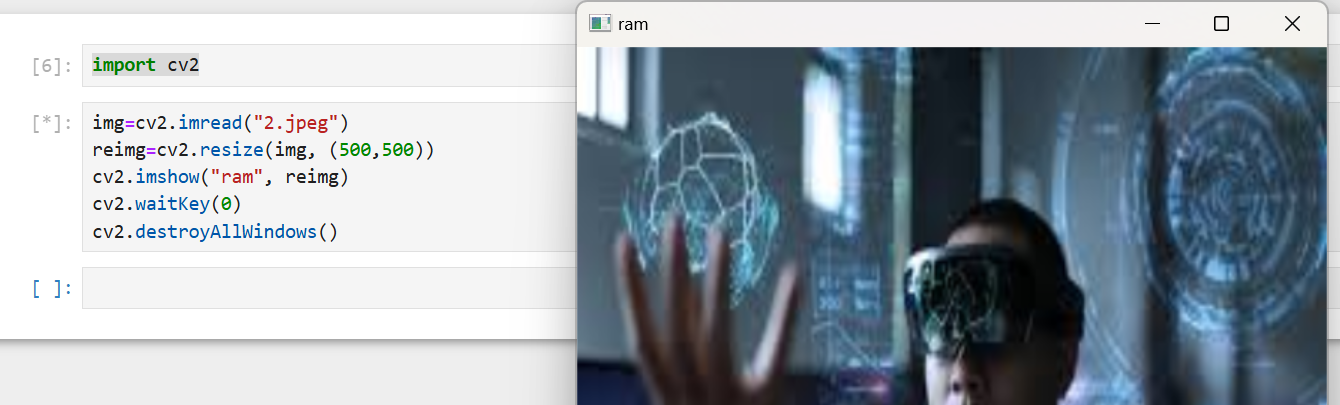
cv2.imshow("ram", reimg)

cv2.waitKey(0)

cv2.destroyAllWindows()

**Input/ Output**

*Input an image after the program executed , image displayed into a resized form*

****

**Result:** *program successfully executed, Image successfully resized and displayed***.**

.

**Experiment -3**

**Experiment 3: Write a program in OpenCV(Using Python) to show multiple images in a single frame.**

***Aim: Using OpenCV and python, show multiple images into a single frame.***

**Algorithm/ Procedure:**

1. Start Program by opening Jupiter Navigator
2. Open Jupiter Notepad
3. Import CV2 library
4. Import numpy variable
5. Insert path of an image to read.
6. Put resize function for resizing image
7. Adjust images horizontally and vertically
8. Use waitKey( ) function for exit
9. Exit

**Program:**

*import cv2*

*import numpy as np*

*img=cv2.imread("SRM.png")*

*reimg=cv2.resize(img, (200, 200))*

*h=np.hstack((reimg, reimg))*

*v=np.vstack((h,h))*

*cv2.imshow("hello", h)*

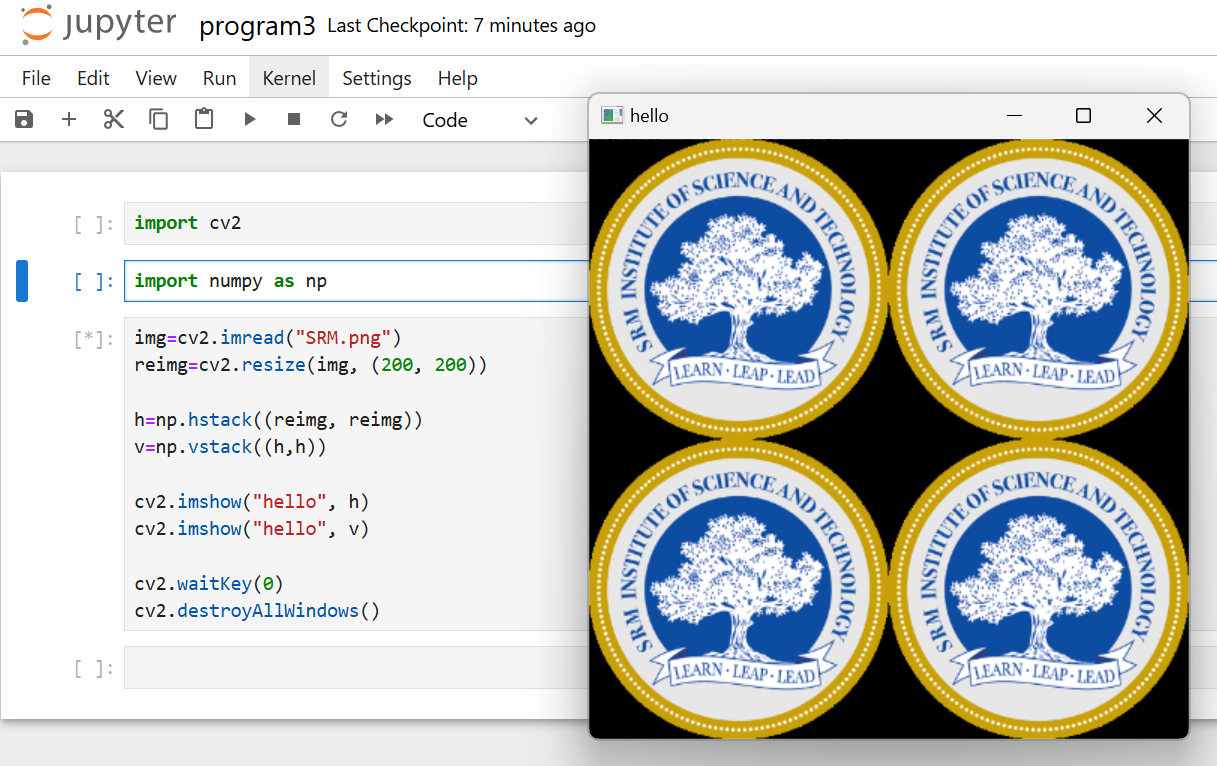
*cv2.imshow("hello", v)*

*cv2.waitKey(0)*

*cv2.destroyAllWindows()*

**Input/Output:**

*Input a figure (SRM Seal) and adjust it into frame 4x4*

**

**Result:** *program successfully executed, image successfully framed.*

**Experiment -4**

**Experiment 4: Write a program in OpenCV(Using Python) crop an image also show its original image.**

***Aim: Using OpenCV and python, crop image from original image.***

**Algorithm/ Procedure:**

1. Start Program by opening Jupiter Navigator
2. Open Jupiter Notepad
3. Import CV2 library
4. Print type of image
5. Print image original size
6. Crop image by using crop variable and ratio
7. Use waitKey( ) function for exit
8. Exit

**Program:**

Import cv2

# Read Input Image

img = cv2.imread("2.jpeg")

***# Check the type of read image***

print(type(img))

print("Shape of the image", img.shape)

**# [rows, columns]**

crop = img[50:180, 100:300]

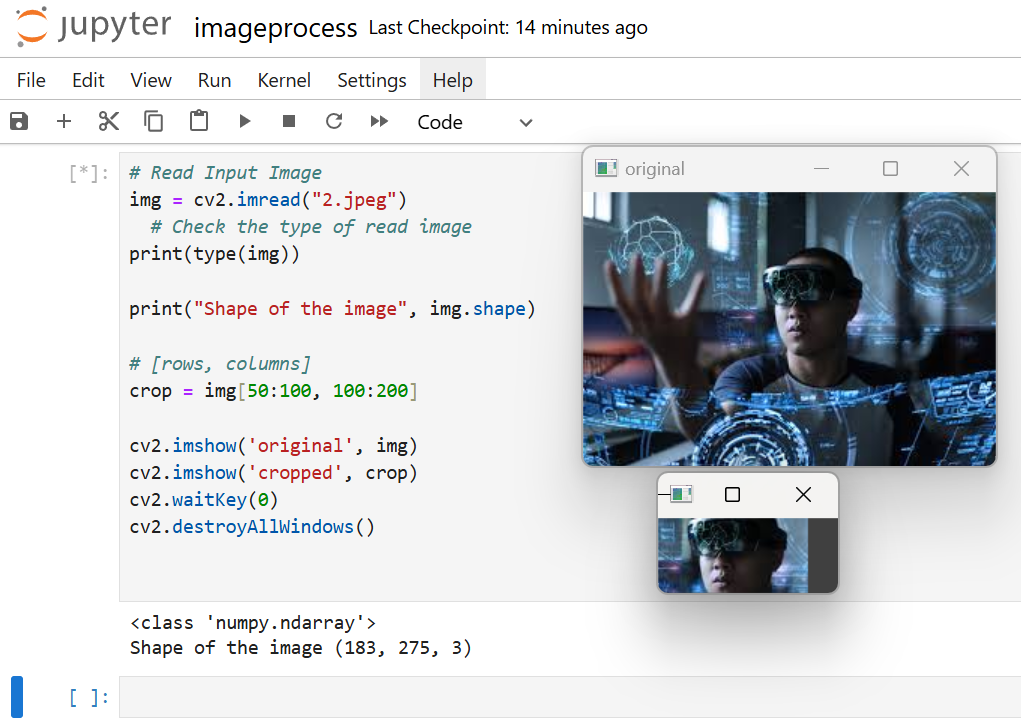
cv2.imshow('original', img)

cv2.imshow('cropped', crop)

cv2.waitKey(0)

cv2.destroyAllWindows()

**Input/output**

****

***Result: Image cropped successfully***

**Experiment -5**

**Experiment 5: Write a program in OpenCV using python to detect edges of an image.**

***Aim: Using OpenCV* to find the edges of an image.**

**Algorithm/ Procedure:**

1. Start Program by opening Jupiter Navigator
2. Open Jupiter Notepad
3. Import cv2
4. Exit

**Program:**

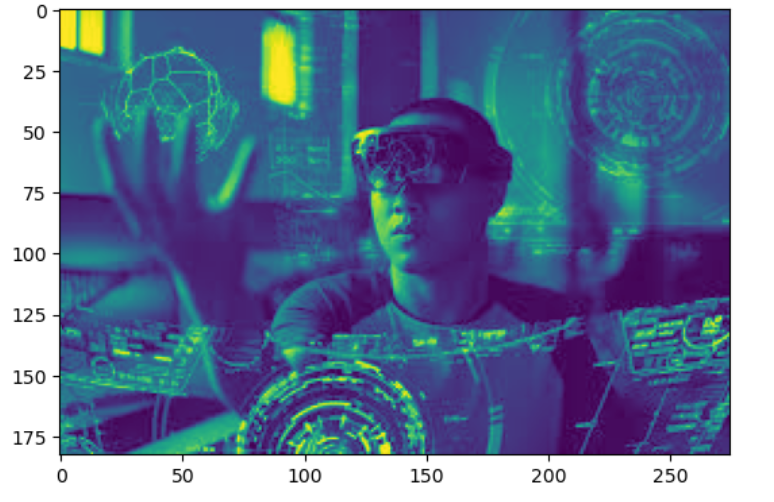
import cv2

import matplotlib.pyplot as plt

img=cv2.imread('ss.jpeg', 0)

plt.imshow(img)

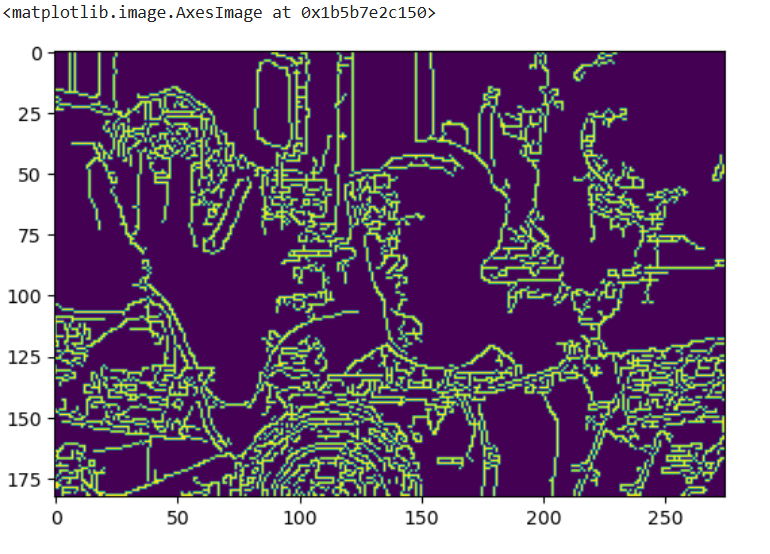
***Input of image with name ss.jpg***

******

***img1=cv2.Canny(img,10,400)***

***plt.imshow(img1)***

***Output:***



**Result:** edges detected successfully

**Experiment -6**

**Experiment 6: Write a program in OpenCV for detecting a face by using AI code.**

***Aim: Using OpenCV* to detect a face using AI cascade.**

**Procedure:**

1. Start Jupyter Notebook
2. Import CV2
3. Import image that is required to detect.
4. Convert image into grayscale
5. Import cascade XML cascade file for detect face.
6. Use for loop for creating square on face
7. Display image
8. End program

**Program/ Code :**

import cv2

import matplotlib.pyplot as plt

# Load the cascade

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

# Read the input image

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

img1=cv2.resize(img, (200,200))

# Convert into grayscale

gray = cv2.cvtColor(img1, cv2.COLOR\_BGR2GRAY)

# Detect faces

faces = face\_cascade.detectMultiScale(img1, 1.2, 5)

# Draw rectangle around the faces

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

cv2.rectangle(img1, (x, y), (x + w, y + h), (255, 0, 0), 2)

# Display the output

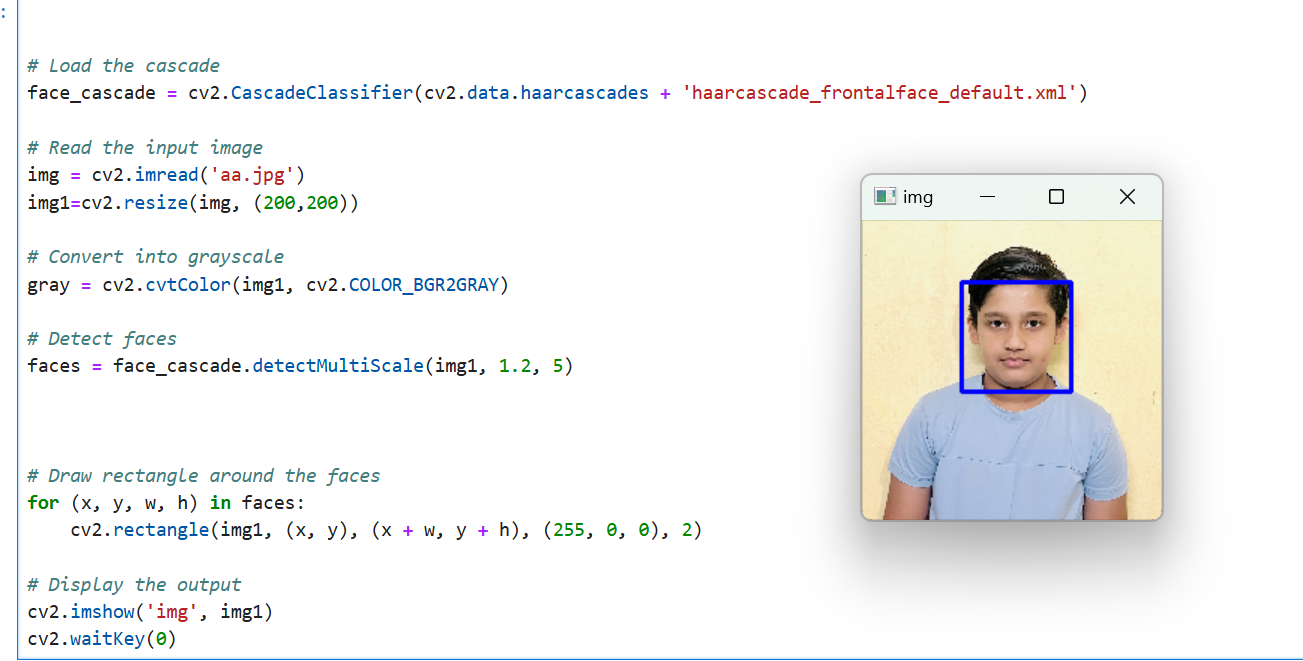
cv2.imshow('img', img1)

cv2.waitKey(0)

**Input / Output**

**Input an image with name aa.jpg**

Output:



**Result:**

Face detected successfully

**Experiment -7**

**Experiment 7: Write a program in OpenCV for detecting eyes on face by using AI code.**

***Aim: Using OpenCV* to detect eyes using AI cascade.**

**Procedure:**

1. Start Jupyter Notebook
2. Import CV2
3. Import image that is required to detect.
4. Convert image into grayscale
5. Import cascade XML cascade file for detect face and eyes.
6. Use for loop for creating square on eyes
7. Display image
8. End program

**Program/ Code :**

import cv2

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

nadia = cv2.imread('./Data/Nadia\_Murad.jpg',0)

def detect\_eyes(img):

face\_img = img.copy()

eyes = eye\_cascade.detectMultiScale(face\_img,scaleFactor=1.2,minNeighbors=5)

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

cv2.rectangle(face\_img, (x,y), (x+w,y+h), (255,255,255), 10)

return face\_img

result = detect\_eyes(nadia)

while True:

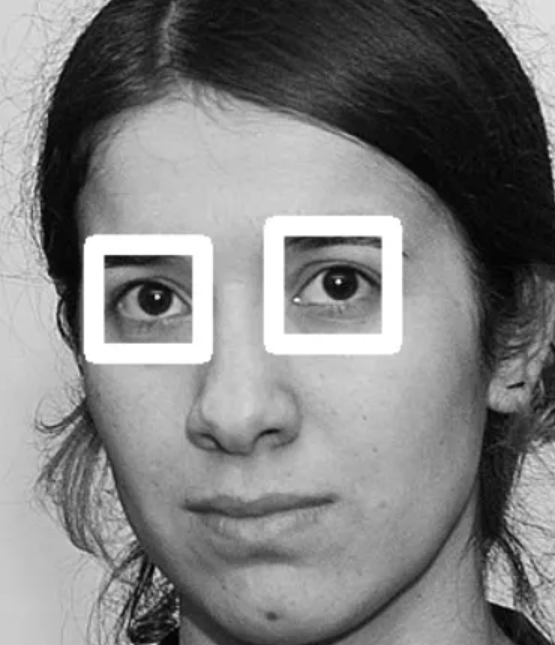
cv2.imshow('',result)

code = cv2.waitKey(10)

if code == ord('q'):

break

**Input / Output**

**Result : Eyes detected successfully by using OpenCV**