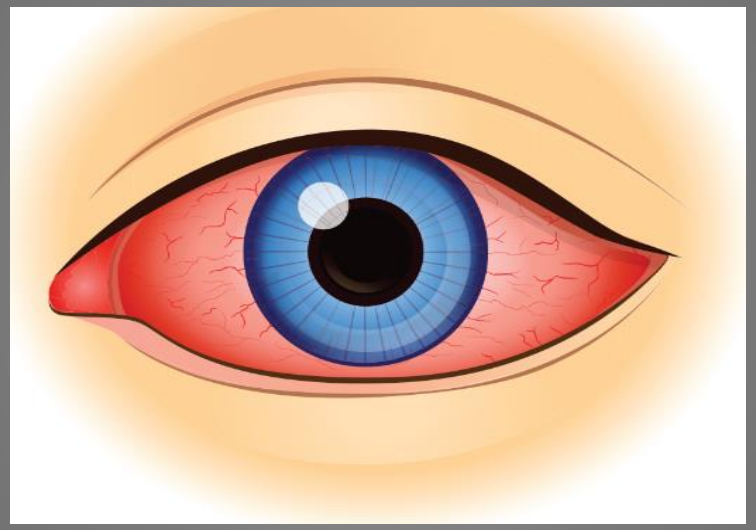
"Rabbi-'shrah li sadri wa yassirli amri wa-hlul 'uqdatan min lisani yafqahu qawli.“

"O My Lord! expand my breast for me. And ease my task for me; and loose a knot from my tongue, (That) they may understand my saying. (Ameen)

OCULAR CONTROLLED SCRIPT DEVELOPER

An Application of Computer Vision and Machine Learning in Medical Diagnostics



Presented By:

Hafiz Rumman Adnan

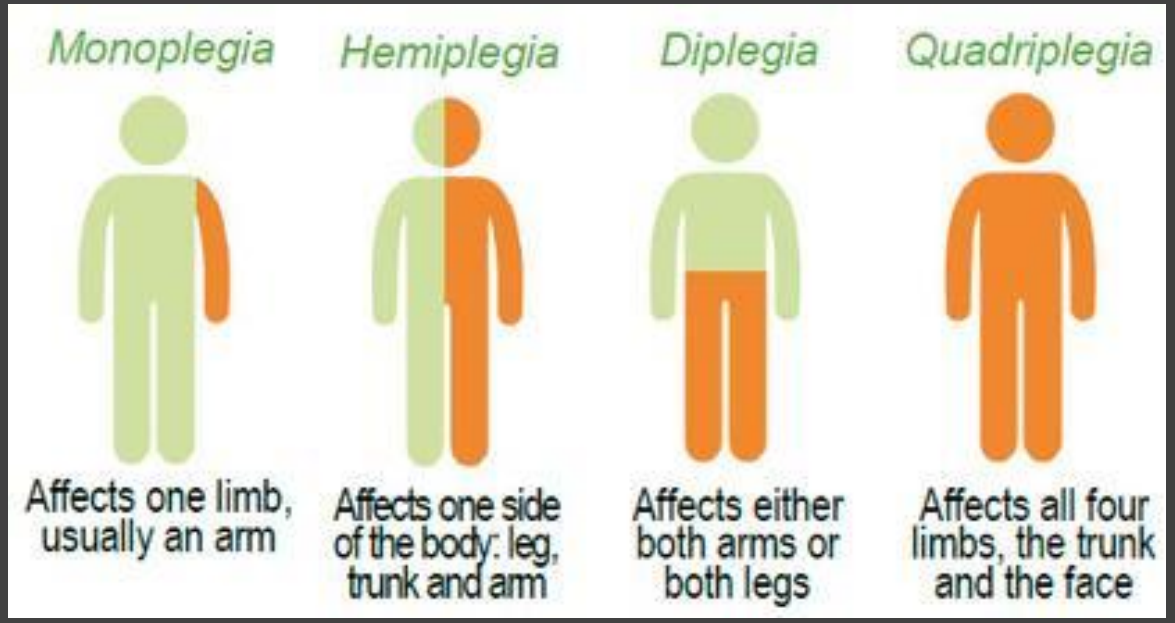
Problem Description

- Paralysis: A worldwide issue
- No affordable eye tracking device available for patients

Types of Paralysis

- Quadriplegia

Basically there are 4 types of paralysis, but we are more concerned with Quadriplegia, in which whole body of the patient seizes to move, only their eyes can move. And they cannot communicate with anyone.



Solution

We are using these modern instruments, software's and hardware's

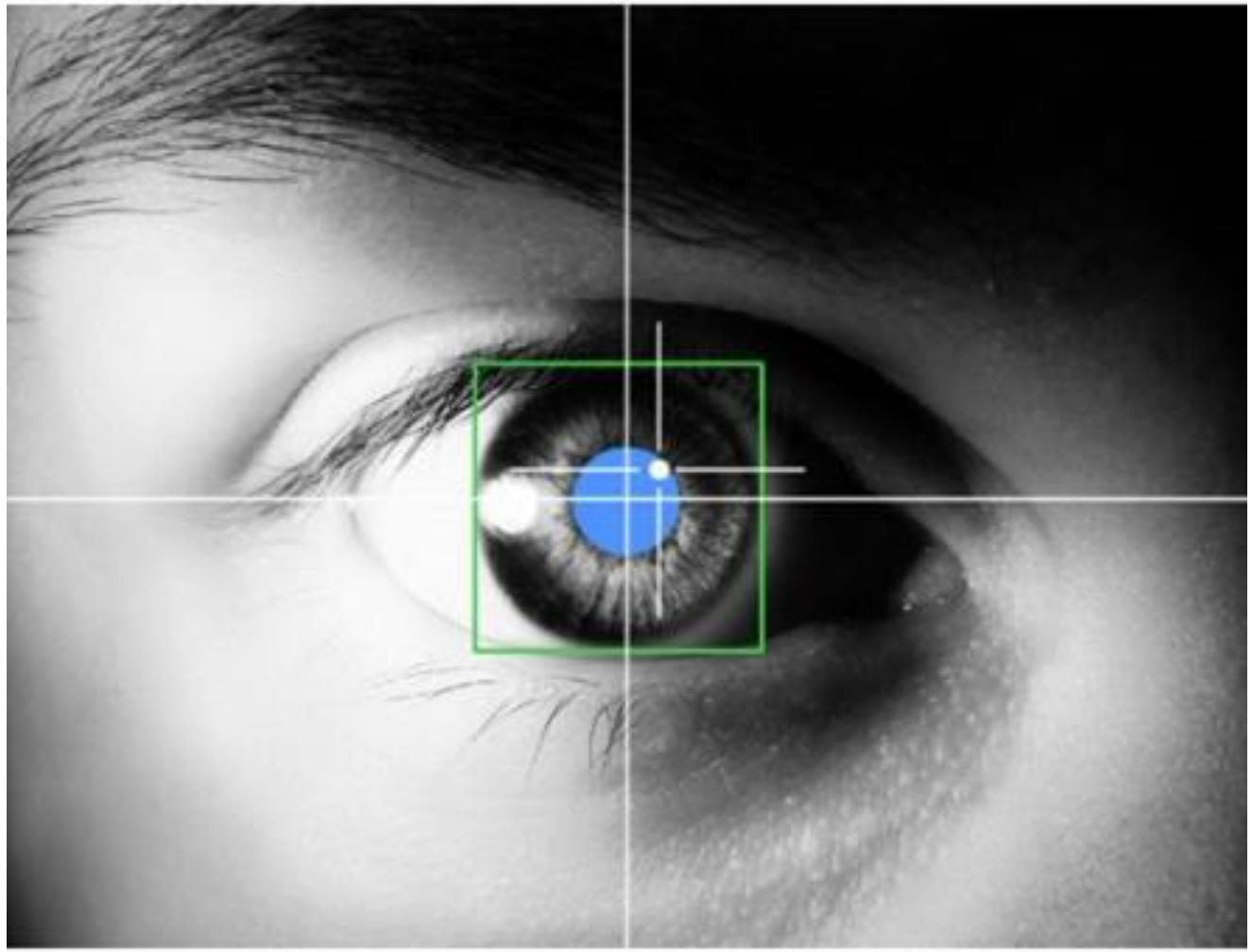
- Microcontroller (Raspberry Pi)
- Programming Skills
- Programming Language PYTHON
- Image Processing Techniques

Ocular Controlled Script Developer

- The Device will allow the patient to write their message on the screen of a computer using only the movement of their eyes.

Basic Idea

- Basic idea of this device is to track the movement of the eyeball and then use it to control the cursor of the computer to write using on screen keyboard.

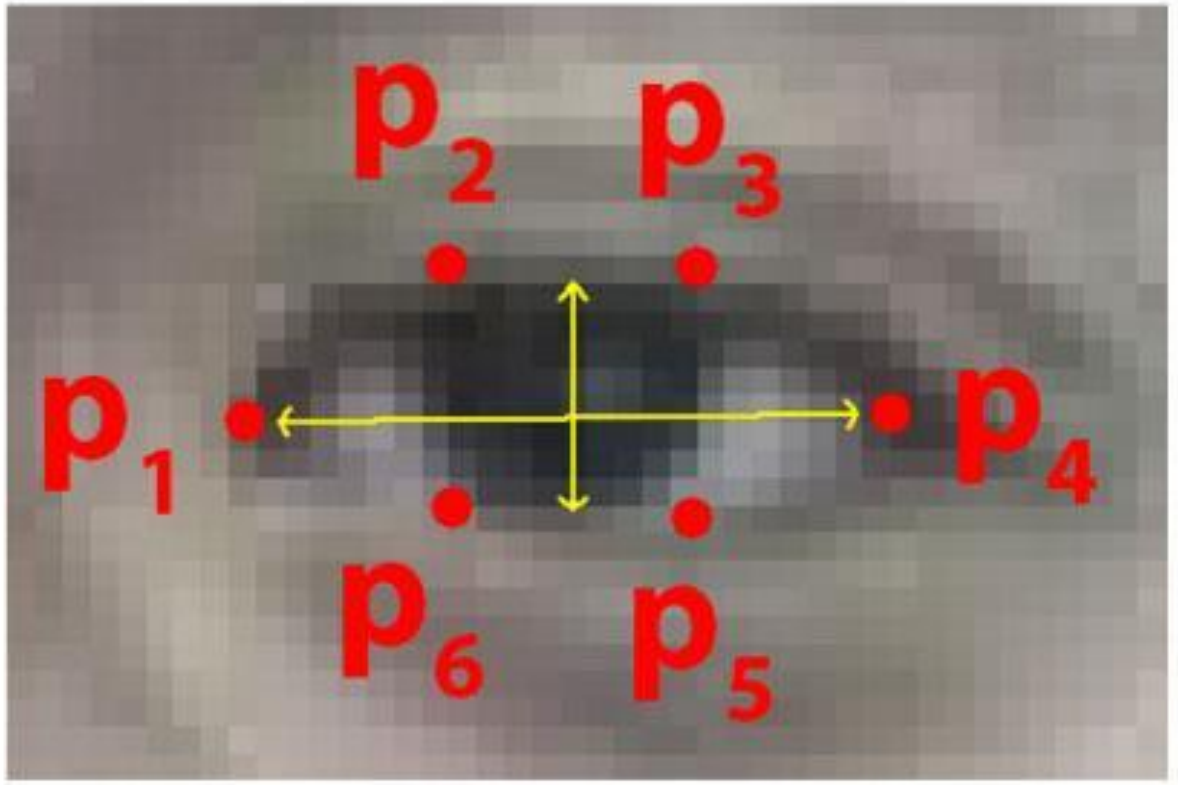


Blink & Wink Detection

- Using Soukopova and cech’s paper published in 2016
- EAR Ratio
- EAR< 0.23
- Blinking (EAR of both eyes)
- Winking (EAR considered separately)
- Left eye wink for left clicking event
- Right eye wink for right eye clicking event

Blink & Wink Detection

$$\mathrm{EAR} = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\|p_1 - p_4\|}$$

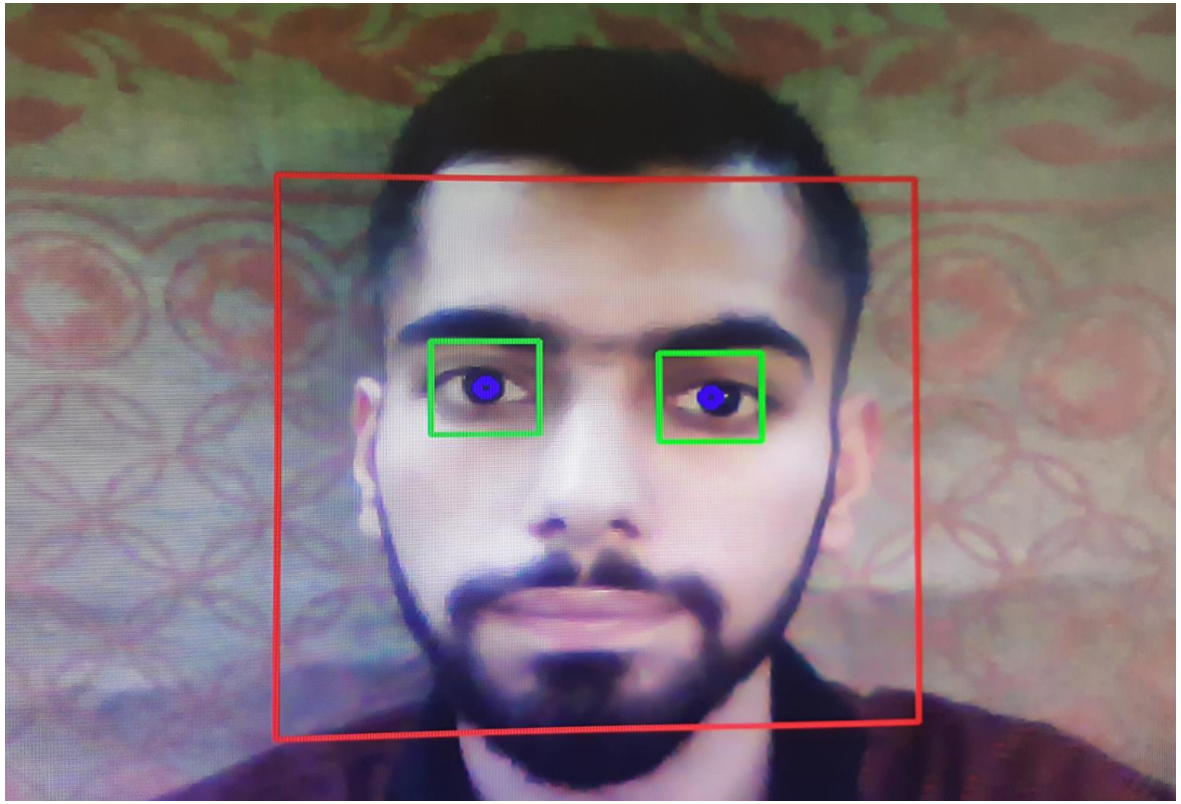


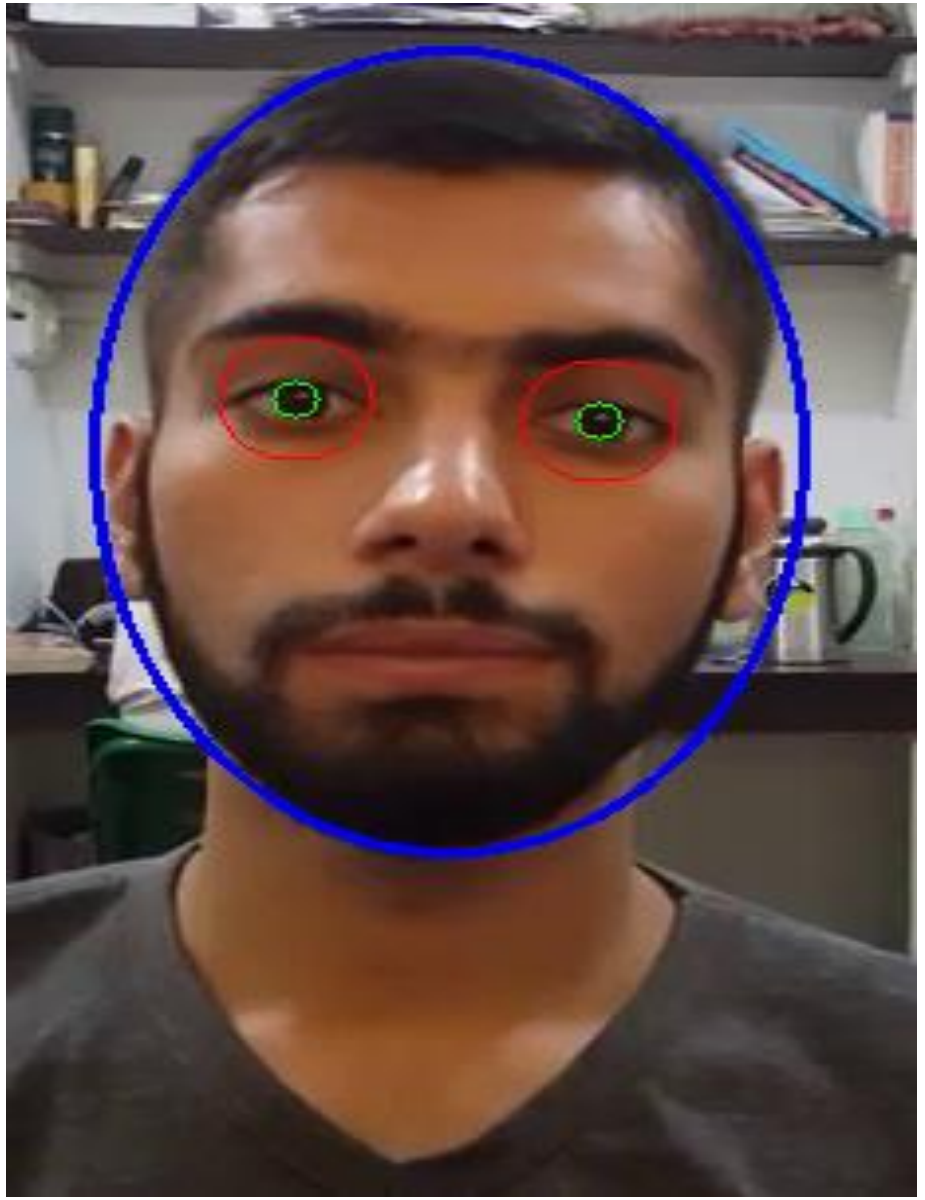
Face & Eye Detection

Pupil Tracking Step:1

- Lbp (local binary patterns) cascade classifier (More FPS)
- Haar Cascade Detector
- Movement from the reference point gives the point of eye movement in next frames.

Changing Different scales and video quality





Reference Point Computation

Step:2

- Centre of Eye work as reference point to control Mouse.
- Compares with pupil center in each frame and give coordinates to mouse function in each frame

```
xc=int(round(ex+ew+ex)/2)
yc=int(round(ey+eh+ey)/2)
cv2.circle(roi_color,(xc,yc),6,(0,255,0),1)
```

Cropping and Zero Padding

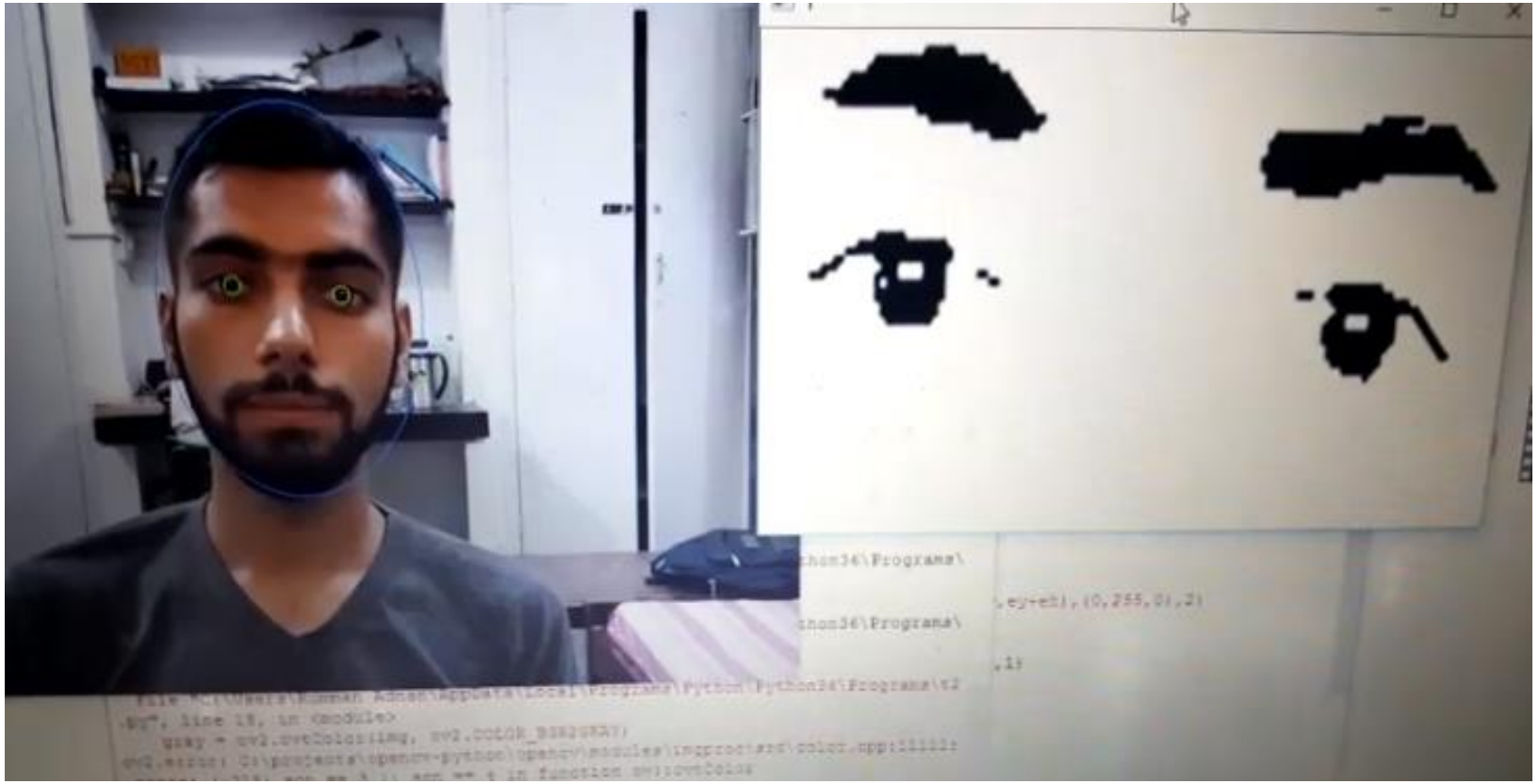
Step:3

- Crop the above part of eyes to see more clearly.
- Padding is done to adjust the eyeballs in center

```
ret,th1 = cv2.threshold(roi_gray_1,40,255,cv2.THRESH_BINARY_INV)
TX = (len(th1[0]))
TY = len(th1)

th1 = np.concatenate((np.zeros((round(TY-TY/8),round(TX))),th1), axis=0)
```

Before



After

Remove Eye lashes

Step:4

- Basic morphology operations
  (Erosion followed by dilation).
- Erosion removes white noise and shrink our object. so we dilate it
- Convolution to sharpen the edges

```
kernel = np.ones((5,5),np.uint8)
opening = cv2.morphologyEx(th1, cv2.MORPH_OPEN, kernel)
  kernel = np.ones((5,5),np.float32)/25
dst = cv2.filter2D(opening,-1,kernel)
blur = cv2.blur(dst,(8,8))
```

After morphologyEx

After Convolution

Blurring & Thresh holding

Step:5

- convolving image with a normalized box filter.

  (takes the average of all the pixels under kernel area and replace the central element)

- Test with different filters but average filter gives best output

- Threshholding is done to find contours in next step

```
.............Step_5 (Blurring & Thresholding).............

blur = cv2.blur(dst,(8,8))
eee,blur1 = cv2.threshold(blur,40,255,cv2.THRESH_BINARY)

blur2 = np.uint8(blur1)
```

Box Filter

$$K = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Results


```
blur = cv2.blur(dst,(8,8))                                  ##  Average filter (Blurring)
eee,blur1 = cv2.threshold(blur,40,255,cv2.THRESH_BINARY)

blur2 = np.uint8(blur1)
```

Finding Contours

Step:6

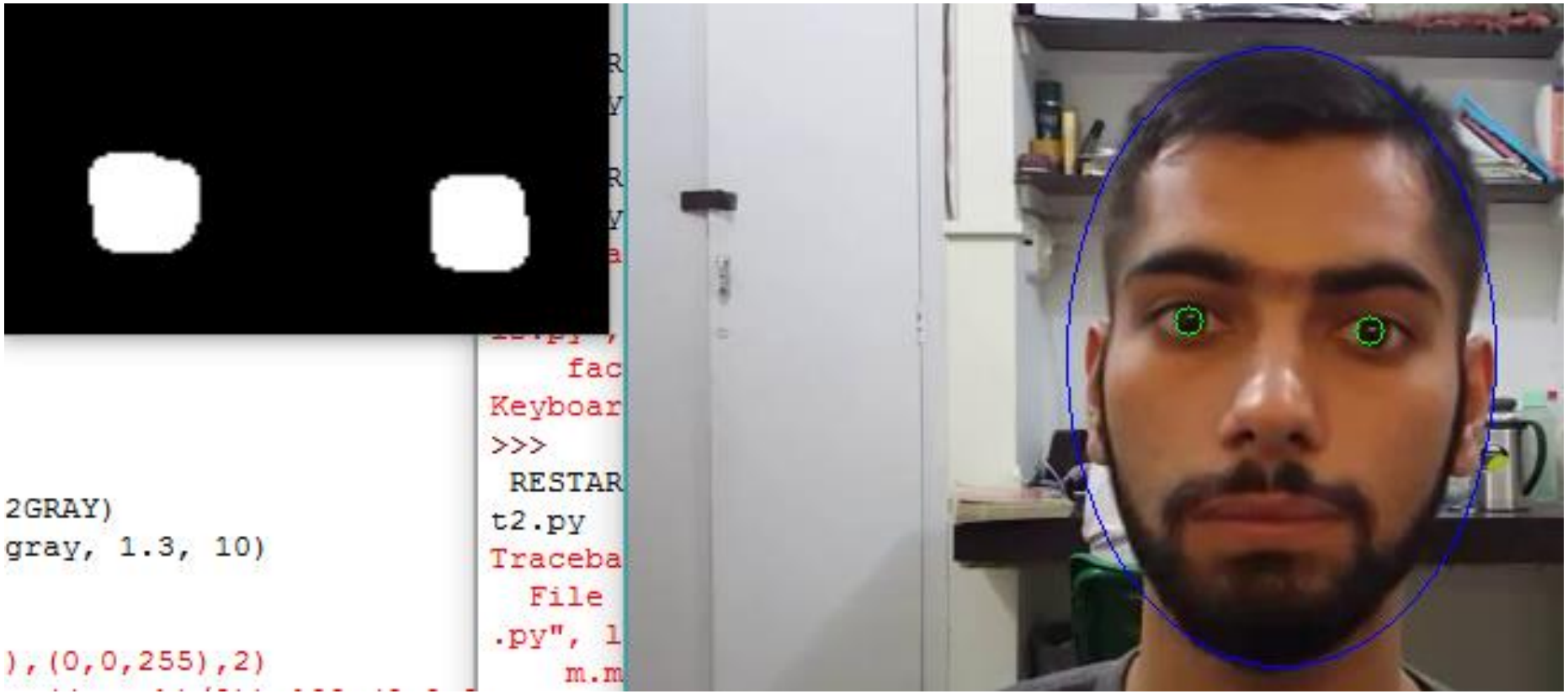
- Contours: curve joining all the continuous points along the boundary.

  (Boundaries of Shapes)

- Stores the (x,y) coordinates of the boundary of a shape using approximation method and draw the boundary of shape.

```
  for c in contours:
compute the center of the contour
            M = cv2.moments(c)
            cX = int(M["m10"] / M["m00"])
            cY = int(M["m01"] / M["m00"])
```

Output: (Centre of contours is center point of pupil)



Mouse Control

Mouse Control

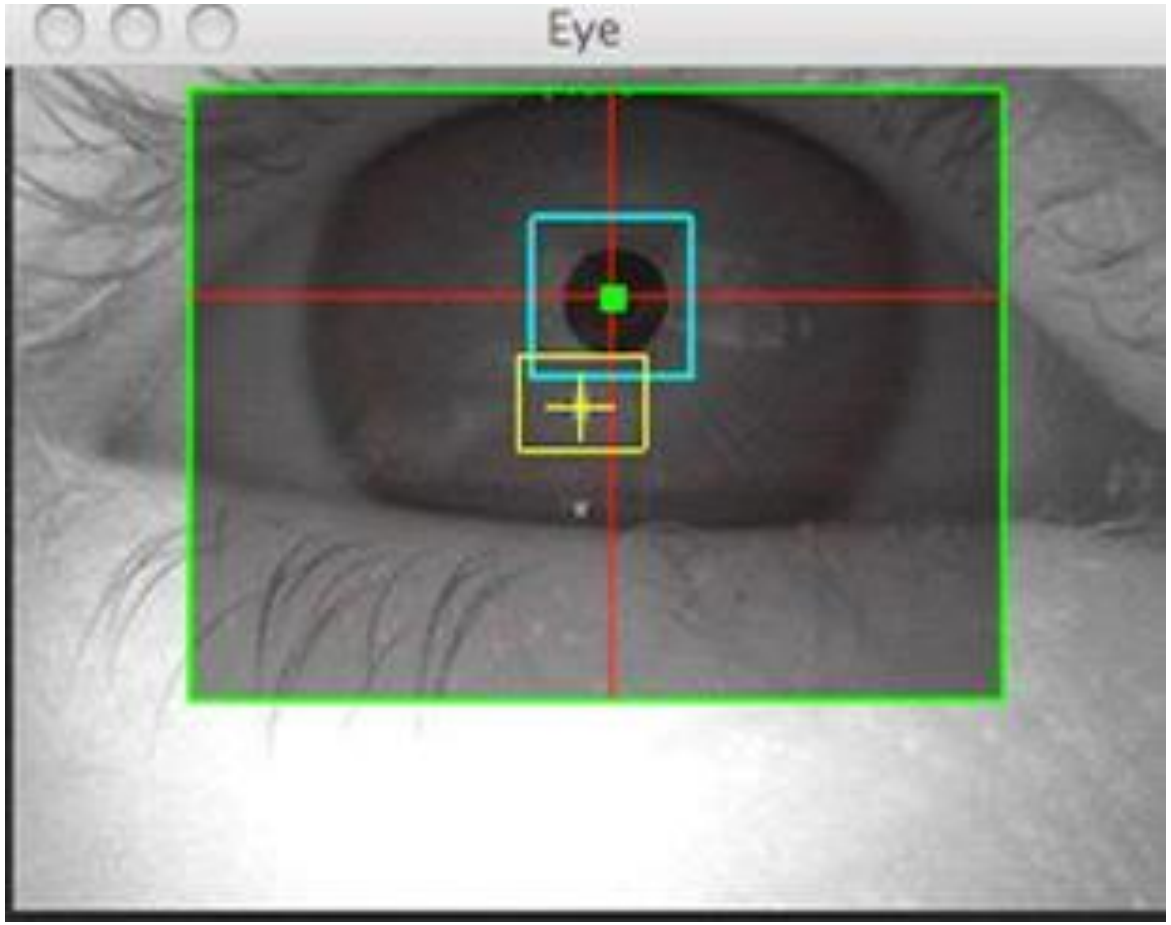
- Centre of Eye work as reference point to control Mouse.
- Compares with pupil center in each frame and give coordinates to mouse function in each frame

```
W, H= pyautogui.size()          ##screenWidth, screenHeight
currentMouseX, currentMouseY = pyautogui.position()
m1=currentMouseX;
m2=currentMouseY;
pyautogui.moveTo(W/2, H/2, duration=1)
```

```
x1=xc-cX
y1=yc-cY
cv2.waitKey(16)
m.movement(x1,y1,1,-2)
```

(xc,yc) reference point, center of eye

(cX,cY) Centre point of pupil tracking



Accomplishments

- Face and Eye Tracking
- Computing Reference point for Mouse Control
- Pupil Tracking in real time
- Wink and blink Detection
- Mouse Control

Future Work

We interact using our eyes for true immersion(deep mental involvement) in games because need to know where we are looking because mostly games need to know where we are looking.

This eye tracking devices can be improved for using in shops. It helps shopkeeper to track products on which customer looks most of the time

It is also used for Multitasking, In cooking your hands are not free but if you want to see the recipe's along cooking you control whole computer eyes

Thank You