JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Focused Based Smart Classroom Management System

Sadhana tiwari

Email-tiwari.sadhana1992@gmail.com

Mobile no.-8871589195

Dr. Manish D Sawale

Email-manish.sawale@gmail.com

Mobile no-9893095600

Abstract: Classrooms are emotional settings. An emotion of a student during a lecture plays a crucial part in any learning climate whether it's in homerooms or e-learning. A key challenge in the teaching process is to identify whether the students can understand the topic clearly or not. To overcome this problem, we use multiple ortho dox models like (tests, quizzes, Assignments, etc) but all these processes depend on multiple factor-like student basics knowledge and intelligence.

We use the eye-based emotion recognition technique to construe significant data to comprehend the state of mind of the students in a classroom environment. Our emphasis is on comprehension and introducing the passionate condition of the student during a learning commitment in a class. Assessing the focus of a student can continuously help in upgrading the learning experience and updating the learning substance. Here, I propose a framework that can recognize and screen of focus a student in a learning climate in a classroom and can give a continuous criticism instrument to improve the learning. This can assist the lecturers in understanding student focus levels. The proposed framework assists with recognizing feelings and identifies the interest of the student in a class.

This can be done by capturing the focus of the students. The focus of a student plays an important role in the classroom or in e-learning. The educator can screen students' interests throughout the class by capturing the images in the student's eyes. This work centres around students in a classroom and in this way comprehend their eye movements which finally helps in improving the quality of education and in arousing interest among the students.

In this research, our focus is to identify the learner's focus. The proposed system can identify boredom in the learning environment and enhance it. The system which is developed will be benefitting all the teachers and will bring a change in the learning environment. The application of this research in a classroom environment is a significant advancement of technology in the field of education. This will change and bring innovation to the field of education and make it simpler for the lecturer or educator to identify his teaching capabilities. Further, it is going to improve the quality of education which is the need of the hour.

Keywords: Eye-Tracking, Emotion Recognition, Image Processing, Machine Learning.

Introduction: Classrooms are emotional settings. Students' emotional experiences can impact on their ability to learn, their engagement in school, and their career choices. Yet too often education research ignores or neutralizes emotions. To improve students' learning and emotional states, reduce teacher burden, and further develop of emotion and learning theories, research efforts should turn to explore how students can learn regardless of their emotional state. We know that some emotions provide a barrier to students' classroom engagement and test performance.

The system is developed for detecting faces and recognizing emotions from an input image. The system is intended to serve as an initial phase in a face-learning system for a desktop application. The resulting system is computationally efficient and able to recognize emotions in faces under moderate changes in lightning, pose and expression. This system is used to understand the state of learning of a student in a classroom. The present learning network center around the vision of persons furthermore, student community working cooperatively towards profound, significant, top notch learning. The accomplishments of computerized correspondence lead learning networks into another measurement. By methods for correspondence advances utilizing various kinds of learning apparatuses, spaces and types of collaboration virtual learning networks are developing for numerous colleges and universities. As the World Wide Web turns into a progressively increasingly famous mechanism for instructional conveyance of separation training[1], the vision of staff and students working cooperatively in a virtual learning network is turning into a reality. The blast of the information age has changed the setting of what is found out and how it is found out to the idea of virtual study halls.

There is an expansion in education around the world. Even in a classroom environment a teacher needs to identify the understanding level of students. The point was to distinguish the degree of understanding appeared by these articulations which causes the teacher to improve their showing style as needs be that keeps the student intrigued and eager during the class. This system encourages the teachers to recognize the understanding of the students in a classroom environment. The subsequent advance suggested that the outward appearances through the activity units (Eyes, Mouth, Eyebrows and Forehead) help the speakers to recognize the contribution and appreciation of the student in the study hall during the talk. The third step suggested that the understudy's demeanors are essentially related to their feelings that thusly recognize their level of appreciation and understanding. The centrality of the investigation was measurably deciphered with the help of a software and supported by a questionnaire which was filled by students in the class and the teachers during the classroom lectures.

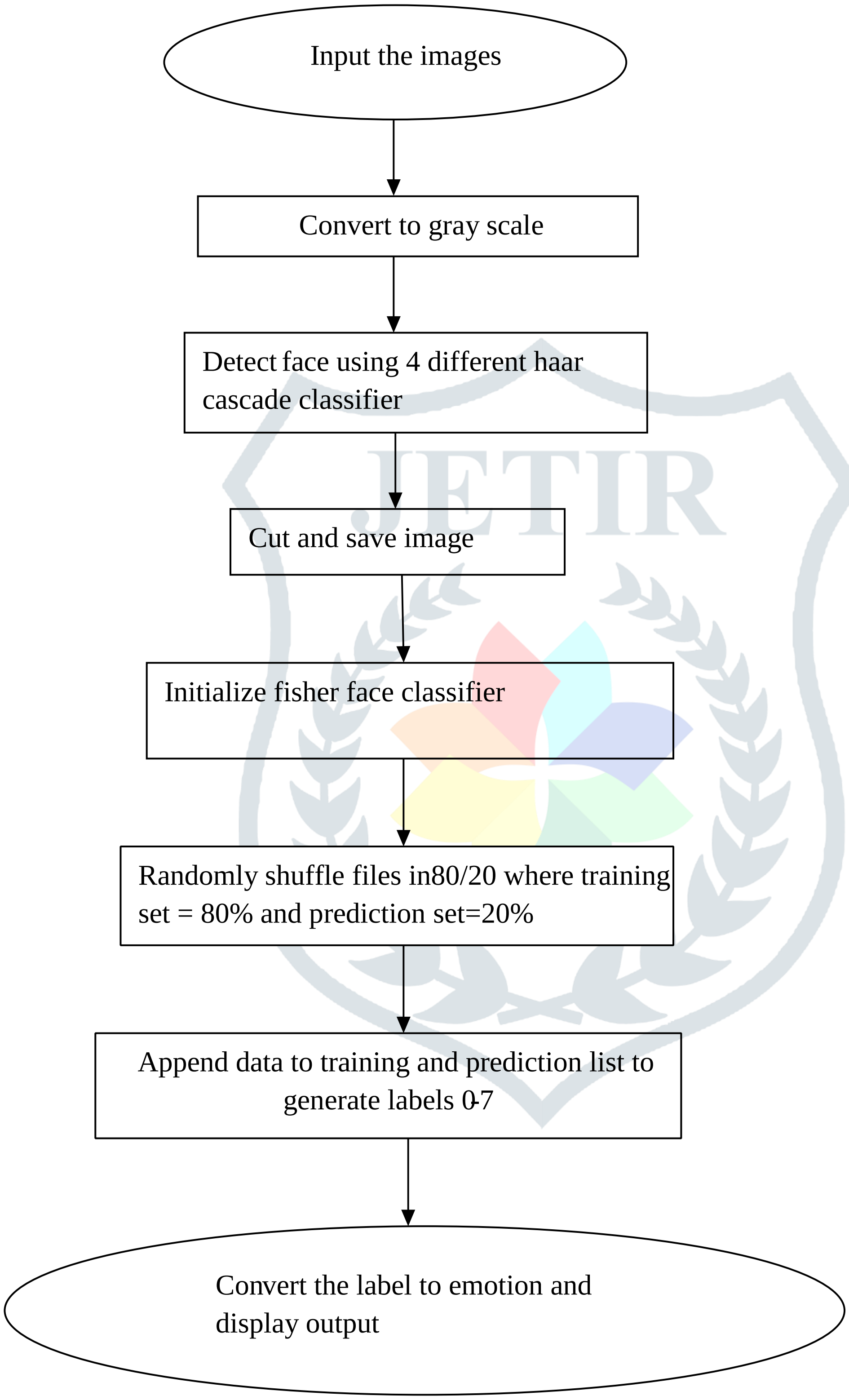
Objectives of the study

- To automatically detect and analyze the learning capabilities of college students.
- To identify the emotions of students using image processing.
- To develop a system that is easy to use, can be easily adaptable, modified, reproduced, and even improved.
- The ultimate goal is to develop a widely acceptable piece of software that will work for the benefit of masses. The user will run the picture in the software and the software which is fully automated will result in the learning capabilities of the student.

Human feelings and goals are communicated through outward appearances and determining a productive and powerful element is the essential segment of facial expression system. Face recognition is significant for the translation of outward appearances in applications, for example, in video conferencing, visual surveillance and the most important communication. Also it is used in real time live motion images. For efficient interaction and communication, facial expressions are the most important. Facial expression recognition is limited to six basic expressions (joy, sad, anger, disgust, fear, surprise) in most research and system. It is discovered that it is insufficient to portray all facial expressions and these facial expressions are categorized based on facial actions. Identifying face, detecting and recognizing facial expression is a quite a very difficult task.

In this study we are identifying the state of learning of students using image processing. The task in this study is accomplished by capturing the images of students in the classroom. These images are fed into a software which identifies the emotion of students. This is further supported by means of two Linkert scale questionnaire where the respondents are both students as well as lecturers. Separate questionnaire is designed for students and lecturers.

Algorithm


Input the images
Convert to gray scale
Detect face using 4 different haar cascade classifier
Cut and save image
Initialize fisher face classifier
Randomly shuffle files in80/20 where training set = 80% and prediction set=20%
Append data to training and prediction list to generate labels 0-7
Convert the label to emotion and display output

Implement

If Personal Computers are to cooperate normally with people, they should understand emotions and express social abilities. In any case, this scenario where a computer or a machine starts to understand our emotions and start interpreting it is in its earliest stages. The chance of keen coaching frameworks that follow understudies' feelings is an appealing idea. Lecturers give individualized instructing in classroom environments. It is easy for a human being to understand the emotions of other people but it's quite a difficult task for a machine. Our exploration objectives are two-overlap: 1) to methodically inspect the relationship(s) between student's feeling state and learning results, i.e., to distinguish whether a relationship exists between students accounted for feelings and their learning of concepts and 2) to assess the significance of students' passionate states an emotional state inside a classroom setting. Subsequently, this investigates how the students' involvement in a classroom is depicted by their emotions or sentiments. It also shows whether a students' sentiments reflect while learning, and whether we are able to understand the learning outcome by emotions. Following are the screenshots of the software.

Uploading image to source folder

Once the image is uploaded the user will receive a message that the image has been uploaded. Here the user will be able to see the image along with its name being displayed on the screen. If the image already exists in the folder, then the user will receive a message that the image already exists and can't be uploaded. In such case the user has to again start the classification with a new image. Otherwise user will click on START CLASSIFICATION button. Next the source data will be organized and the user needs to click on ADD TO DATASET button to add the image to the dataset. Once the image is added to the dataset, user will receive a message that the image is added to the dataset. Then the user needs to click on CHECK ACCURACY button. Once the user clicks on CHECK ACCURACY button, he will be able to see the accuracy of the detection algorithm. Also the user can click on NEXT button to view the results of the classification.

The user will be able to see the buttons of each emotion and he can click on VIEW PHOTOS to view the images of the corresponding emotion.

Eye Detecting

Figure: Image to detect eye

Real-time eye tracking using OpenCV and Dlib

Real-time eye tracking we will need a facial keypoints detector that can detect eyes in real-time. For this will use a pre-trained network in the dlib library which can detect '68 key points' that was presented in this paper. The required pre-trained model can be downloaded from here. Dlib is used because it can give predictions in real-time.

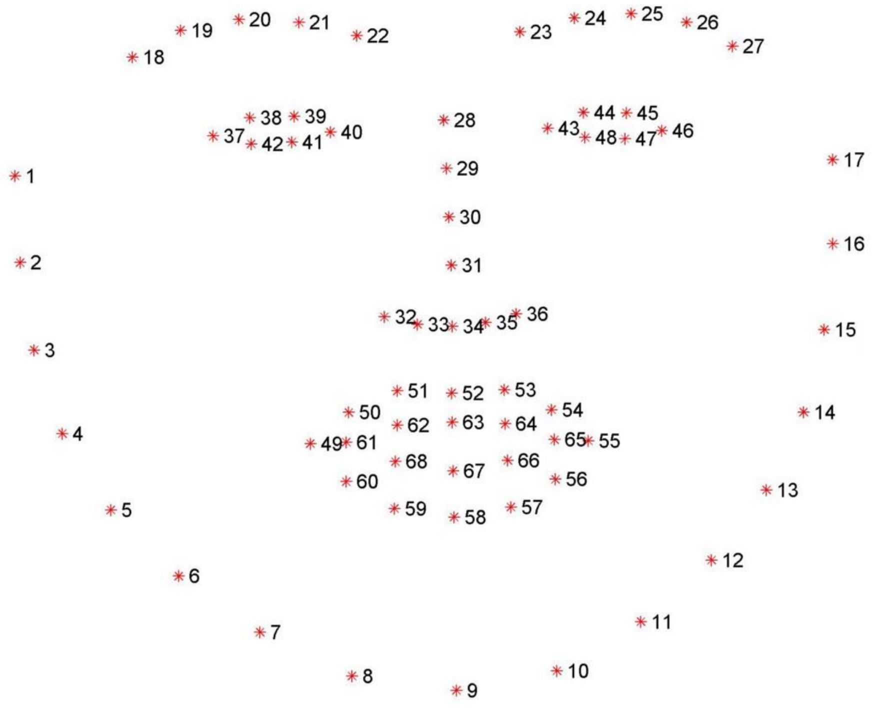


Figure Dlib facial keypoints

The first thing to do is to find eyes before we can move on to image processing and to find the eyes we need to find a face. The facial keypoint detector takes a rectangular object of the dlib module as input which is simply the coordinates of a face. To find faces we can use the inbuilt frontal face detector of dlib. You can use any classifier for this task.

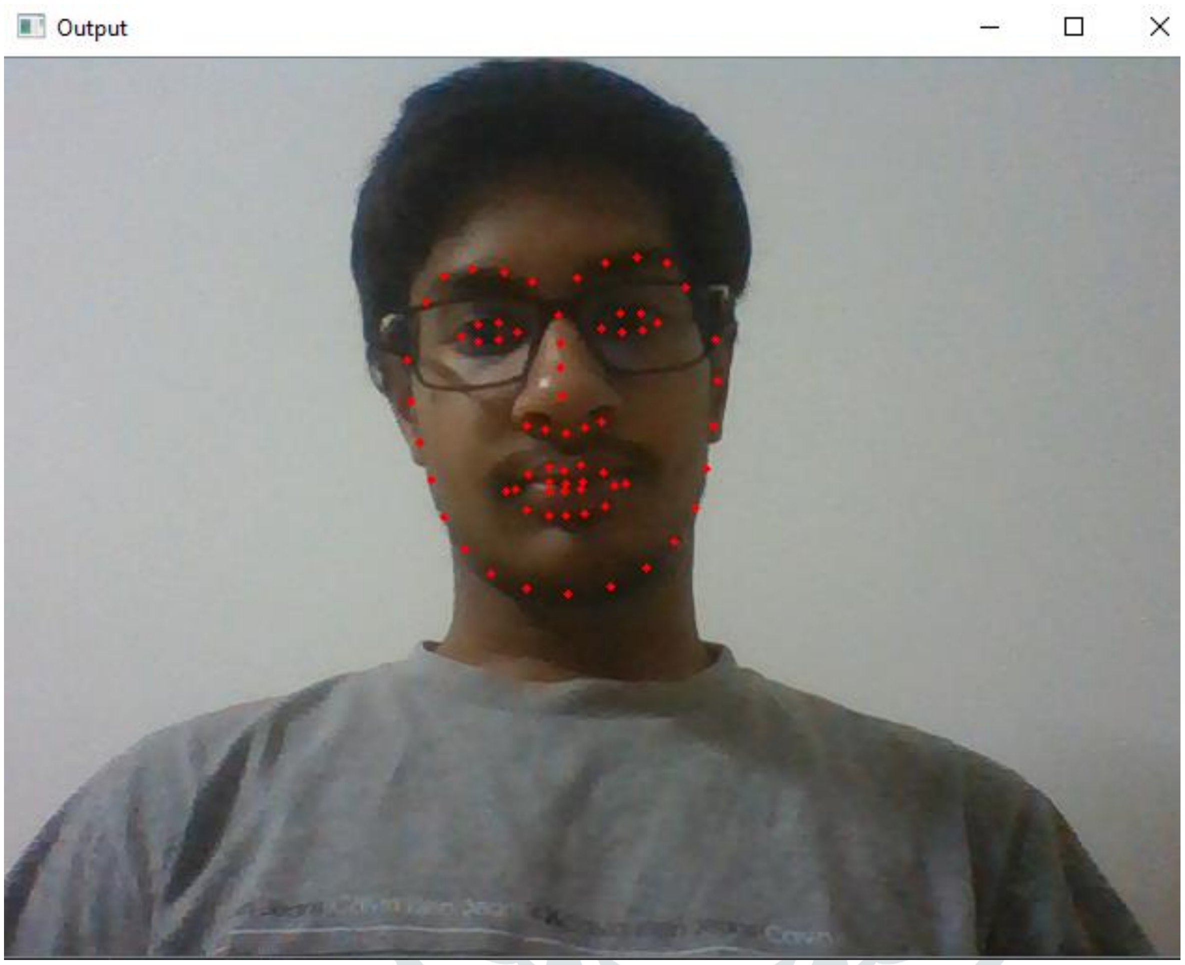


Figure Output Obtained

Finding the center of eyeballs using opencv. We will be working with the live feed obtained through webcam. To open the webcam, read frames, and display it you can use the code shown below:

```
import cv2cap = cv2.VideoCapture(0)
while(True)
  ret, img = cap.read()
  cv2.imshow("Output", img)
  if cv2.waitKey(1) & 0xFF == ord('q'): # escape when q is pressed
    break
```

So now we how to read frames from webcam, it's time to work on them to reach our goal. We create a new black mask using NumPy of the same dimensions as our webcam frame. Store the (x, y) coordinates of the points of the left and right eyes from the keypoint array *shape* and draw them on the mask using cv2.fillConvexPoly. It takes an image, points as a NumPy array with data type = np.int32 and color as arguments and returns an image with the area between those points filled with that color.

```
def eye_on_mask(mask, side):
    points = [shape[i] for i in side]
    points = np.array(points, dtype=np.int32)
    mask = cv2.fillConvexPoly(mask, points, 255)
    return maskleft = [36, 37, 38, 39, 40, 41] # keypoint indices for left eye
right = [42, 43, 44, 45, 46, 47] # keypoint indices for right eye
mask = np.zeros(img.shape[:2], dtype=np.uint8)
mask = eye_on_mask(mask, left)
mask = eye_on_mask(mask, right)
```

After doing this we have a black mask where the eye area is drawn in white. This white area is expanded a little using a morphological operation cv2.dilate. Using cv2.bitwise_and with our mask as the mask on our image, we can segment out the eyes. Convert all the (0, 0, 0) pixels to (255, 255, 255) so that only the eyeball is the only dark part left. Convert the result to grayscale to make the image ready for thresholding.

```
kernel = np.ones((9, 9), np.uint8)
mask = cv2.dilate(mask, kernel, 5)
eyes = cv2.bitwise_and(img, img, mask=mask)
mask = (eyes == [0, 0, 0]).all(axis=2)
eyes[mask] = [255, 255, 255]
eyes_gray = cv2.cvtColor(eyes, cv2.COLOR_BGR2GRAY)
```

Thresholding is used to create a binary mask. So our task is to find an optimal threshold value against which we can segment out the eyeballs from the rest of the eye and then we need to find its center. But the threshold value will be different for different lighting conditions so we can make an adjustable trackbar for controlling the threshold value.

```
def nothing(x):
    pass
cv2.namedWindow('image')
cv2.createTrackbar('threshold', 'image', 0, 255, nothing)
threshold = cv2.getTrackbarPos('threshold', 'image')
_, thresh = cv2.threshold(eyes_gray, threshold, 255, cv2.THRESH_BINARY)
thresh = cv2.erode(thresh, None, iterations=2)
thresh = cv2.dilate(thresh, None, iterations=4)
thresh = cv2.medianBlur(thresh, 3)
```

We have reached the final step of our project. The eyeballs are segmented out and we can utilize cv2.findContours for finding them. Right now our background is white and our eyeballs are in black. However, in OpenCV's cv2.findContours() method, the object to find should be in white and the background is black. So we need to invert our *thresh* using cv2.bitwise_not. Now we can find contours. Theoretically, we can say that all we need to do is now find the two largest contours and those should be our eyeballs. However, this leaves out a little room for false positives that can be tackled by finding the midpoint between the eyes and dividing the image by that. Then we find the largest contours in those divisions and should be our eyeballs. The key points 40 and 43 (39 and 42 in Python because index starts from zero) are used to find the midpoint. Find the largest contours on both sides of the midpoint by sorting it with cv2.contourArea. We can utilize cv2.moments to find the centers of the eyeballs.

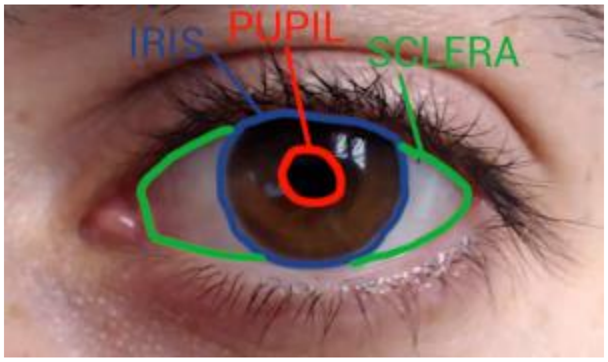
```
def contouring(thresh, mid, img, right=False):
    cnts, _ = cv2.findContours(thresh, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_NONE)
    cnt = max(cnts, key = cv2.contourArea) # finding contour with #maximum area
    M = cv2.moments(cnt)
    cx = int(M['m10']/M['m00'])
    cy = int(M['m01']/M['m00'])
    if right:
        cx += mid # Adding value of mid to x coordinate of centre of #right eye to adjust for dividing into two parts
    cv2.circle(img, (cx, cy), 4, (0, 0, 255), 2)# drawing over #eyeball with redmid = (shape[39][0] + shape[42][0]) // 2
contouring(thresh[:, 0:mid], mid, img)
contouring(thresh[:, mid:], mid, img, True)
```

On running this our code throws several errors like max() arg is an empty sequence or division by zero which are thrown when no contour is found or M['m00'] is zero respectively. To solve this, enclose the contouring function in a try block. We don't need to do anything if an error is thrown which will occur only when eyes are not detected. The new contouring function will look like:

```
def contouring(thresh, mid, img, right=False):
    cnts, _ = cv2.findContours(thresh, cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_NONE)
    try:
        cnt = max(cnts, key = cv2.contourArea)
        M = cv2.moments(cnt)
        cx = int(M['m10']/M['m00'])
        cy = int(M['m01']/M['m00'])
        if right:
            cx += mid
        cv2.circle(img, (cx, cy), 4, (0, 0, 255), 2)
    except:
        pass
```

Study of Eye

Before getting into details about image processing, let’s study a bit the eye and let’s think what are the possible solutions to do this.
In the picture below we see an eye.



The eye is composed of three main parts:

- **Pupil** – the black circle in the middle
- **Iris** – the bigger circle that can have different color for different people
- **Sclera** – it’s always white

Let’s now write the code of the first part, where we import the video where the eye is moving. And later on we will think about the solution to track the movement.

We import the libraries Opencv and numpy, we load the video “eye_recording.flv” and then we put it in a loop so tha we can loop through the frames of the video and process image by image.

```
import cv2

import numpy as np

cap = cv2.VideoCapture("eye_recording.flv")

while True:

ret, frame = cap.read()

if ret is False:

break
```

Let’s now **select an Roi (region of interest)**. In this way we are restricting the detection only to the pupil, iris and sclera and cutting out all the unnecessary things like eyelashes and the area surrounding the eye.

```
roi = frame[269: 795, 537: 1416]
```

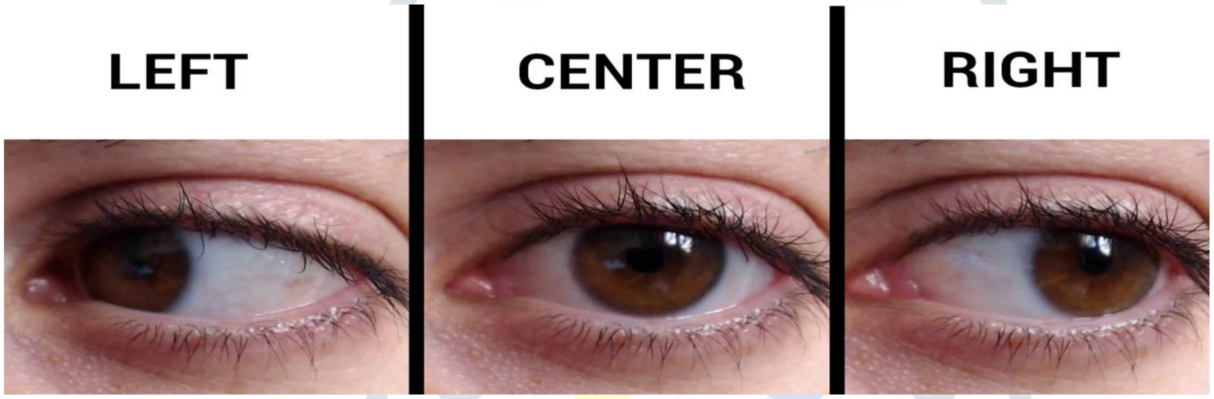
```
rows, cols, _ = roi.shape

gray_roi = cv2.cvtColor(roi, cv2.COLOR_BGR2GRAY)

gray_roi = cv2.GaussianBlur(gray_roi, (7, 7), 0)
```

Now we can dive deeper into finding the right approach for the detection of the motion.

Let’s take a look at all possible directions (in the picture below) that the eye can have and let’s find the common and uncommon elements between them all.



What can we understand from this image?

Starting from the left we see that the sclera cover the opposite side of where the pupil and iris are pointing. When the eye is looking straight the sclera is well balanced on left and right side.

DETECTING THE MOTION

For the detection we could use different approaches, focusing on the sclera, the iris or thepupil. We’re going for the easiest approach possible, and probably the best solution anyway.

We **will simply focus on the pupil**. By converting the image into grayscale format we will see that **the pupil is always darker then the rest of the eye**. No matter where the eye is looking at and no matter what color is the sclera of the person.

So let’s do this. First conversion to grayscale and then we find the threshold to extract only the pupil.

```
gray_roi = cv2.cvtColor(roi, cv2.COLOR_BGR2GRAY)

gray_roi = cv2.GaussianBlur(gray_roi, (7, 7), 0)
```

```
_, threshold = cv2.threshold  (gray_roi, 3, 255, cv2.THRESH_BINARY_INV)
```

From the threshold we find the contours. And we simply remove all the noise selecting the element with the biggest area (which is supposed to be the pupil) and skip al the rest.

Result

We did our study in two ways – through questionnaire and by fisher face algorithm capturing the images. The fisherface algorithm was found to be 62.5% accurate. We found a correlation coefficient value of 0.799 between students and their expressions and the focus of student and teacher correlation coefficient value was 0.698. We find variations in fisherface algorithm and focus detection is 0.101. This shows that students face normal in some other reasons.

Emotion recognition system frameworks are improving continuously much over the previous decade and a lot of research is going on in this regard. Face Detection and Extraction of emotions from facial pictures is helpful in numerous applications.

We will simply focus on the pupil. By converting the image into the grayscale format we will see that the pupil is always darker than the rest of the eye. No matter where the eye is looking and no matter what color is the sclera of the person.

References

[1] B.Woolf et. al.,"Affect-aware tutors:recognising and responding to student affect," In Proc. of Int'l J. Learning Technology, vol. 4, pp. 129-164, 2009.

[2] A. C. Amit Konar, Emotion Recognition: A Pattern Analysis Approach, Wiley, March 2015.

[3] A. G. a. A. R. S. L. Happy, "A real time facial expression classification system using Local Binary Patterns," in 4th International Conference on Intelligent Human Computer Interaction (IHCI), 2012.

[4] A. K. R. a. H. MOHAPATRA, "Fundamentals of Software Engineering: Designed to Provide an Insight Into the Software Engineering Concepts (English Edition)".

[5] A. m. Ma Ralph, "Image Processing pipeline for facial expression recognition under variable lighting".