

Major Project Report
on

**RECOGNISING STUDENTS’
ATTENTION IN SMART CLASSROOM**

Submitted in partial fulfillment of the requirement for the award of the degree of

**Bachelor of Technology
in
Computer Science and Engineering**

Under the supervision of:

Mrs. Leena Singh
Assistant Professor
Dept. of CSE
Amity School of Engg. and Tech.

Submitted by:

Aditya Sahu, 00210402716
Rahul Joshi, 02610402716
Chirag Arora, 41210402716
Ankit Singh, 00510402716



Department of Computer Science and Engineering

AMITY SCHOOL OF ENGINEERING AND TECHNOLOGY

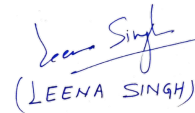
(Affiliated to Guru Gobind Singh Indraprastha University, New Delhi)

[2016-2020]

CERTIFICATE

It is hereby certified that the project entitled “AI Smart Classroom” has been submitted by Aditya Sahu (00210402716), Ankit Singh Tanwar (00510402716), Rahul Joshi (02610402716) and Chirag Arora (41210402716) of CSE 8th Semester, under my guidance as a part of B.Tech major project.

This work put in by them is an outcome of their own hard work and effort and the matter embodied in the report has not been submitted for the award of any other degree.



(LEENA SINGH)

Mrs. Leena Singh
Assistant Professor
Department of Computer Science & Engg.
Amity School of Engineering and Technology, New Delhi

Date: 21st July, 2020

ACKNOWLEDGEMENT

In the course of development of this project many people have helped us on various levels. First of all, we would like to thank **Prof. Dr. Rekha Agarwal**, Director and **Dr. Pinki Nayak**, Head of Department (Department of CSE/IT) for their constant encouragement and guidance throughout the project, thus enabling us to perform our best. We would also like to thank our guide **Mrs. Leena Singh** for her unbound technical guidance and ideas that have helped us enrich and make this project better at each level. Her cordial support, valuable information and guidance, helped us in completing this task through various stages. We express our sincere thanks to the entire faculty of Amity School of Engineering and Technology for giving us all the facilities during our training period.



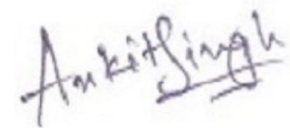
Aditya Sahu



Rahul Joshi



Chirag Arora



Ankit Singh

LIST OF FIGURES

Figure No.	Title	Page No.
Figure 1.1	System for Smart Classroom	03
Figure 1.2	Classroom setup	05
Figure 1.3	Database to Flutter architecture	07
Figure 2.1	Shift in the head pose of students	12
Figure 3.1	Design of the proposed system	14
Figure 3.2	Haar Cascade Features	17
Figure 3.3	LBPH operation on a single pixel.	18
Figure 3.4	Histogram extraction of the image.	19
Figure 3.5	Head pose estimation to find the angles - yaw, pitch and roll	20
Figure 3.6	Schema of the database	22
Figure 3.7	Fetches data into the app	23
Figure 4.1	Facial recognition of many students in classroom	24
Figure 4.2	Unknown person in frame	25
Figure 4.3	Head pose estimation	25

LIST OF TABLES

Table No.	Title	Page No.
Table 4.1	Comparison between different existing systems and the proposed system	26

ABSTRACT

Smart class for future that this proposed system put forward will significantly enhance the coherent communication and learning exposure among teachers and students using machine learning in the real-time sensing environment. Taking into consideration, the past and current growth in machine learning, it brought us to critical landmarks, that is, it can be used as a component and source of an envisioned smart class. In this report, we majorly focused on three components for a smart classroom system. The in-class computer system that is developed in this report is capable to make real-time suggestion so as to improve the memorability and quality of the lecture it gives and make corrections/adjustments accordingly in real-time. This also includes the primary method to evaluate the attention of students automatically during the lecture in the classroom. The existing research in the field of machine learning based emotion recognition, affect sensing and real-time mobile cloud computing is the base for this proposed system. There are various deep learning algorithms to train the classifiers to estimate the dependent level of attention level each student. Even though the approaches and technologies used in this system are advanced enough to understand and handle the tasks, some challenges lies in merging all of these advanced technologies together, perform these in real-time, and decide the valid and absolute educational parameters that is used in these algorithms. In this report, how deployment of the system is emerged from current scenario issues and give direction to engineering and educational disciplines for the future.

TABLE OF CONTENTS

CERTIFICATE	i
ACKNOWLEDGEMENT	ii
LIST OF FIGURES	iii
LIST OF TABLES	iv
ABSTRACT	v
TABLE OF CONTENTS	vi
1. Introduction	01
1.1. Overview	01
1.2. Student Engagement and Attention in the Classroom	01
1.3. Automated Measurement of Effective Parameters	04
1.4. Human estimation of attention level of students	04
1.5. Attention Estimation Classifiers	05
1.6. Database connectivity to the cloud	06
1.7. User Interface	06
1.8. REST Services	07
2. Existing and Proposed System	08
2.1. Traditional Existing System	08
2.2. Related Works	08
2.2.1. Methods of Data Collection	09
2.3. Discrepancies in Existing System	10
2.4. Proposed System	11
2.4.1. Parameters for determination of engagement level	11
2.4.2. Features of the proposed system	12

2.4.3. Data collection	13
2.5. Objectives	13
3. Design & Implementation	14
3.1. Design of the Proposed System	14
3.2. Implementation	15
3.2.1. OpenCV	15
3.2.2. NumPy	16
3.2.3. Haar Cascade	16
3.2.4. Local Binary Pattern Histogram (LBPH)	17
3.2.5. Head Pose Estimation	19
3.2.6. Integration API	21
3.2.7. Database MongoDB	21
3.2.8. Mobile Application (Flutter)	22
4. Results and Discussion	24
4.1 Results	24
4.2 Discussion	26
5. Conclusion and Future scope	27
5.1. Conclusion	27
5.2. Future scope	28
References	29
Appendix	30

CHAPTER 1 - INTRODUCTION

1.1 Overview

An automated learning approach became an essential part of the educational community, requiring essential systems to keep track of the learning processes and provide response to the teachers with the use of application. Past advancements in visual sensors and computer visualization have allowed automatic inspection of behaviors of the students at various levels of education. Learner variables such as happiness, confusion, fatigue, etc. They are determined automatically by the face and attention level calculated from a variety of visual features.

The proposed system provides a mobile app that provides the presenter with student attention levels in the classroom. It improves the quality of presenters' teaching approach that helps us make the change in their non-verbal behavior.

The first and most important issue was to describe the student's attention in a way that was in line with the teacher's observations. The scores that people's attention spans are analyzed and compared to visual behaviors, activities, gestures and other activities that students display. Those results have been used to explain the sense of monitorable levels of student behavior. The next issue was to select and find sensible features that could accurately differentiate among various attention levels.

The final issue was to find appropriate machine learning methods, capable of learning the standard attention model, applicable to all students or other known person in the classroom.

1.2 Student involvement and attention in the classroom

Student engagement remained a vital topic within academic literature since the late 20th century. The initial enthusiasm for engagement was driven by concerns about huge numbers of student dropouts and statistics showing that several students, estimated between 25% and

60%, were reportedly bored and expelled from the classroom [1, 2]. Such statistics have identified academic institutions to treat the topic of student's engagement not as a goal of improving the performance grades but as yet another independent goal [3]. Nowadays, promoting student engagement became an important aspect not only in the traditional classrooms but also in other forms of learning such as teaching systems (ITS) and Massively Open Online Courses (MOOCs). Some academic research communities have developed numerous taxis to explain student involvement.

In case of higher academics, the measurement of students' involvement in the learning process is especially needed to improve learning outcomes and to assess lessons. This is often done with questions, but with the advent of modern learning strategies, it is now easy to have comprehensive usage of data for the involvement of students in the learning processes. A review of research to measure student involvement in technology-based education provided a review of measures of assessment and quantitative (tools) to evaluate behavioral, cognitive, and emotional indicators of student engagement. Attention has been categorized as one of the aspects of psychological involvement, while fear, stigma and blandness contribute to emotional engagement.

The learning process is much needed to analyze learning and improve learning outcomes in higher education programs. This analysis was done with a variety of questions, and yet with the rapid growth of e-learning, it became much easier to gather data and measure activity and student engagement. The study reviewed participatory and micro-cognitive approaches to student engagement in technology-based learning programs to measure students' cognitive and emotional indicators. Figure 1.1 shows a general idea of a master class system that incorporates machine learning techniques and feedback control.

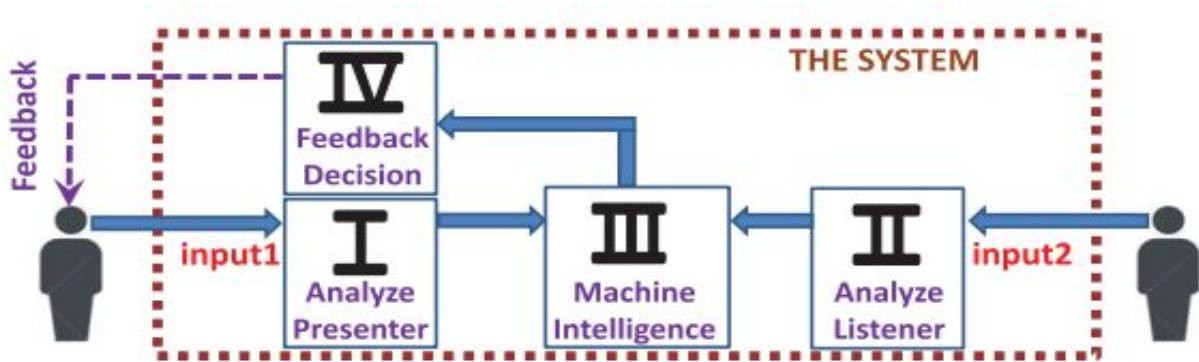


Fig. 1.1: System for Smart Classroom

Attention is defined as “the cognitive process of selectively focusing on a discrete aspect of information, ignoring the comprehensible information”. In various educational settings, terms such as continuous focus or alertness are used to describe a student's ability to maintain long-term focus, as in classroom lectures. Pedagogical research tends to focus on keeping the student's attention in every lecture, because continuous focus is gained as an important aspect of the learning process. Visual viewing by human viewers is an approach that is inaccessible, and real-time video recordings with coding can be used for manual recognition; but due to the limited performance of this method, long-term monitoring should be performed using automated computer detection techniques.

Neural networks, Support vector machine (SVM), k-NN etc. are space-based methods for the recognition of facial expressions in classification algorithms. There are others beyond studying Time and Space based methods such as “a regression neural network”, “a hidden Markov model (HMM)”, “a spatial and temporal motion energy templates method”. The discussion here will be on the efficiency of the classification algorithm for speech recognition and the feature vector used. There is a problem with the feature vector size as its distances can quickly change the size of a single full image to multiple image sizes together. Although the complexity of the classification of this algorithm can be reduced when the vector size of the element is reduced, the actual difference depends on the explanatory ability of the element. Therefore, the most memorable part of this report includes this reduction in feature vector size while maintaining high recognition accuracy. It is not a good practice to assume that the face

of a topic has already been acquired prior to the recognition and removal processes even though much research on the face of doing so. The above was the first issue here but as discussed in detail in this report the main challenge would be to integrate all features and modules as a whole-time system.

1.3 Automated Measurement of Effective Parameters

Video recorded signal (RGB) is often used for passive visual input during guessing of active parameters, such as measuring engagement of the student from facial expressions. Video tests and facial analysis often require high-quality hardware to increase the visibility of many students, which makes them expensive but increases the accuracy and complexity of image analysis in a class-room setup. Techniques for measuring headaches are very useful for integrating functional compasses such as focusing on computer learning environments.

1.4 Human Estimation of Attention Level of Students

Case studies have no explanation how to classify the students' attention in a classroom setting. Readers' reviews of video recording during lectures have shown that their attention to the subject is shown by certain visual behaviors such as writing, observing, and imitation. Through the results of the research, human observers have noted that the attentional dimensions are not always flexible and consistent, the points of understanding indicate temporal fluctuations. Media filtering is done to make the predictions more timely, with a 10s time clock and three levels of flow. The latter reference provides a human estimate of the level of student recognition of the 3 levels and the student's attention is recognized. Figure 1.2 showing a typical classroom setup with a webcam interaction of a student's face and a screen that displays mood scales.

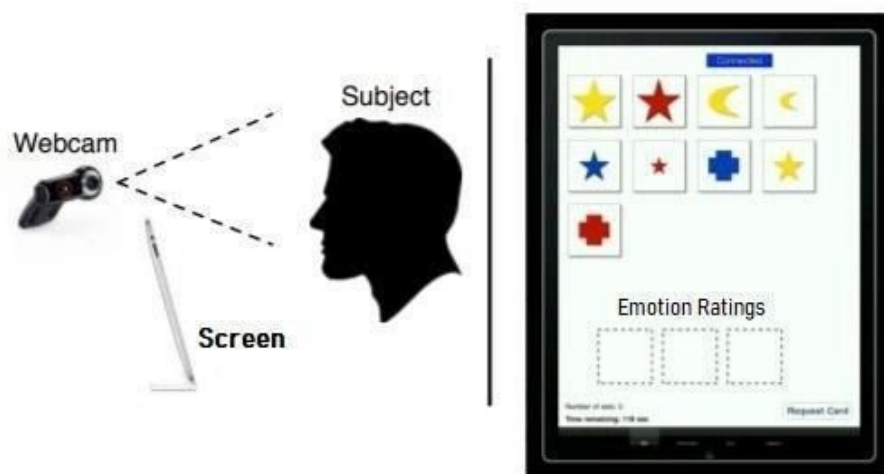


Fig. 1.2: Classroom setup

1.5 Attention Estimation Classifiers

This study thus far has the goal of building an attention classifier to automate the process of estimating 3-dimensional attention from the Kinect features recorded. Choosing the appropriate classifiers and its parameters was the first problem here in order to construct a model for the attention of the average person without looking at the details of the training.

The choice of a complete correction of the features to obtain the most accessible accuracy of the predicted data was a secondary problem. General assessment and testing tested for student-designed features. To prevent overloading of the model, the training process used is five-factor verification.

1. The classifiers examined and their parameters are as follows:
2. Decision Tree (simple): Largest values of sponges is four.
3. Decision Tree (average): Largest values of spines is twenty.

4. K-nearest neighbors (coarse): Euclidean is the estimated distance; the number of neighbors is 100 and the weight of the distance is the same.
5. Neighbors near K : Measurement distance is Euclidean; the number of neighbors is 10 and the weight of the distance is moderately averaged.
6. Decision tree fund: The Ensemble method is a fund; The type of student is the decision maker.

1.6 Database connectivity to the cloud

The output/data generated from each of the modules is transported to the mongodb online database cloud which helps in delocalization of the data which will be further used in the application made. For this connectivity mongodb atlas cloud has been used which helps to form clusters of the data received from the scripts provided and performs database administration tasks automatically like configuring the database, provisioning the infrastructure. Mongodb Atlas cloud uses a unique userID and password for each data cluster which helps in providing privileges such as admin, read/write access or read-only access.

MongoDB is a document-directed cross platform database program. It is categorized as a NoSQL database program. It will be used to store the daily activity of each student and their attentivity scores. It will be used by the REST API services to perform write, read, and delete operations.

1.7 User Interface

The system is supposed to be user friendly in order for teachers to easily work along with it. For this reason development of a mobile application is taken into consideration as it will be feasible for any teacher to just look up in the mobile device to monitor the students. There are mainly two platforms used by mobile devices these days that are android and iOS so the

application should be able to work on either of them. This can be done using any native mobile SDK (software development kit) software like ionic apps, flutter apps etc.

In this project Flutter is used because its response time is very fast as compared to other SDKs as well as it is famous for its beautiful and attractive frontend.

1.8 REST Services

The proposed system uses Representational state transfer (REST) architecture which involves the third element i.e REST API(Representational State Transfer Application Program Interface Services).

These services are used by the proposed system to perform READ, WRITE, UPDATE, DELETE operations. REST APIs directly communicate with the database as described in section 1.6 because the user interface cannot directly do so. The services are written in Node.js. The three elements of the architecture are shown in figure 1.3.

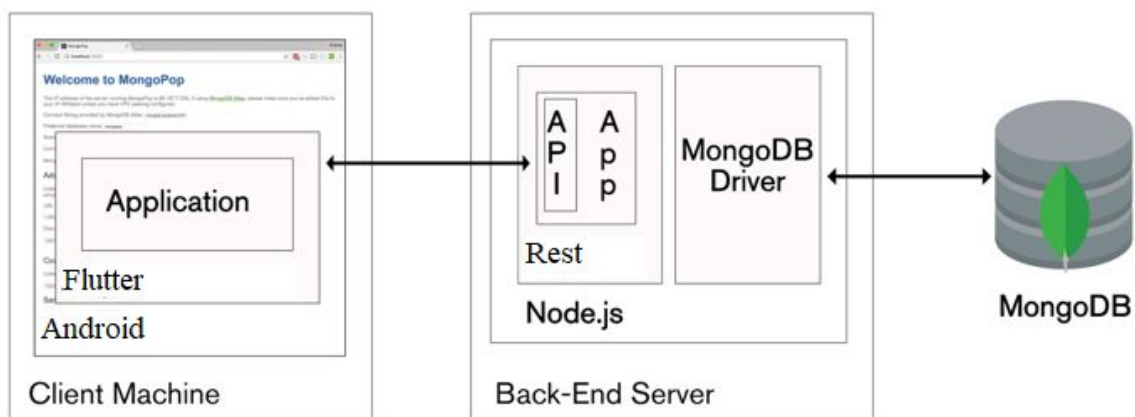


Fig. 1.3: Database to Flutter architecture

CHAPTER 2 - EXISTING AND PROPOSED SYSTEM

2.1 Traditional Existing System

Student involvement is the main aspect of modern education, and is seen as a goal in its own way. The existing system comprises a traditional classroom within which the teacher stands at the front teaching the students who are seated facing the blackboard. There is minimalistic use of technology for analysing the student's behaviour which is the main reason that the teacher has to maintain the discipline of the class by monitoring every student's activity and checking their attentiveness while teaching simultaneously. This is obviously neither practical nor feasible to do so, and the issue gets even worse as the number of students in the class increases.

In the traditional system, what happens is that the teacher grades the students based on their memory of how each student performs. It is simply not correct to conclude that the teacher remembers history about how much engaged is each student in their class. Moreover, human misjudgement can also take place sometimes. Due to these reasons, the system suffers some massive drawbacks and failures which leave the students unjustified even when they were attentive and fully engaged in the classroom. The entire system relies upon the judgement made by the teacher, whose truthfulness is not guaranteed.

It is, therefore, required to make use of some technology in the classroom which assesses the students based on their engagement and behaviour in the class and record it in a persistent database.

2.2 Related Works

Zaletelj et al. [4] in various ways measured the students' level of attention in a classroom setup, one of the methods is "using a set of features calculated from the data obtained by the Kinect One sensor". The Kinect sensor detects facial features through computer vision

algorithms, such as mouth, eyes and nose and enters a wide 3D model. On the basis of visual perception they had acquired a body set, staring and facial features that were related to the students' behavioral characteristics and their relation to the level of attention measured by human spectators. However, sometimes adoption can fail and may have serious consequences. This is due to the reliability of eye detection based on the presence of object block and face direction (forward or not). Also, Kinect sensors do not explain all the variations in the visual behavior of test people from standard data included in a set of seven signals.

Whitehill et al. [5] examined “the real-time automatic recognition of engagement from students' facial expressions”. A dataset of students' facial features was collected during comprehension training work. Their results have shown that machine learning techniques can be used to perform automatic detectors with better precision compared to human monitors. They show that personal and spontaneous decision-making are in line with the task. However, it faces problems using it literally. Most students are accustomed to participating, which is different from the prolonged engagement or miscommunication seen in classrooms. This leads to the importance of longitudinal studies that mimic the environment in a classroom where some of the students are assuring and others are unfairly dismissed.

Behnagh et al. [6] explored the practical and technological components of the AI class that is emotionally active for the future. Their program provides “real-time automated feedback by using two modal attributes in the teacher to improve speech efficiency - teacher self-regulation and metacognitive awareness, and their non-verbal communication skills”. The program uses advanced techniques for deep learning, multimodal sensors, GPU computing, and mood detection and visual data to retrieve hand gestures, voice interpretations, and body language information of the presenter. On the other hand, the program receives credits from the students to determine the presentations.

2.2.1 Methods of Data Collection

Whitehill et al. [5] collected “data from 34 graduate students who participated in the "Cognitive Skills Training" survey conducted in 2010-1-1”. The aim of this assessment was to

calculate the importance of teaching by seeing the student's face during a lecture or presentation. They collect video performance data and their work in courses that work with software to learn comprehension skills. Tested they used their software built for perceptual skills and installed it on the Apple iPad. The webcam was placed immediately behind the iPad, was directly on the student's face and was used to record students' feeds in the video.

In a scheme suggested by Zaletelj et al. [4], “the actual phase of the extraction feature captured the video and orthogonal data obtained during the experiment and recorded it on a disk drive”. Thereafter, Matlab documents are used to process and analyze the information received. Several forms of data are provided by the Kinect SDK body and facial recognition engine are released through the recording system. The colored frames are digitized at frame rates “up to 15 frames per second with a resolution of 1920 X 1080 pixels”. The frames are encoded and saved as an “H.264 video file”. The frame depth of the forecast for attention is written with a resolution of 512 X 424 pixels that can be used.

2.3 Discrepancies in Existing System

In the traditional system, the teacher has to pay attention to every student in order to analyze their attention and behaviour and that too while teaching. This can be a very hectic job for one person to perform and is less efficient. From students part, they find it sometimes difficult to easily interact with the teacher during lecture as not to create disturbance. This is not the fault of the teacher as one can go off the track easily when stopped in between presenting detailed information.

Even after the evolution of modern classroom systems which do not consider most of the important parameters required to measure the engagement level of students in the classroom. The system proposed by Zaletelj et al. [1], Whitehill et al. [2] take into consideration only the facial features as their only parameter to judge whether the student is attentive or not, which is not only insufficient, but might lead to inaccurate results as other parameters such as head

pose estimation, bodily gestures, expressions, etc. also contribute to the engagement level exhibited by the students.

2.4 Proposed System

The main purpose of the system proposed is to determine the engagement of each student in the classroom by capturing their facial expressions, gestures and activities and storing them in the database. The face of the student needs to be captured in such a manner that all the features of their face needs to be detected, even the seating and the posture of the student need to be recognized.

2.4.1 Parameters for determination of engagement level

The proposed system defines some important parameters which judge the engagement levels of students just as how a human teacher would analyze. These parameters were established by studying and examining various samples of real classes and seeing how students react in the presence of the teacher during the lecture.

- A. Physical presence
- B. Head pose estimation

Physical presence: The physical presence of the student will determine the engagement level of the student i.e whether the students attend the whole lecture or not.

Head pose estimation: When observed from where the blackboard/screen is, the students are usually facing directly opposite to the position of the teacher. Often, the students bend their head down to read or make notes. This means that the angle of the head of the students should be either close to facing front, or bent down to the bottom. As shown in Figure 2.1, the student might learn backward or turn to their colleagues to discuss something related to the class — most of the time when the lecture is going on, they are likely to face on the two sides mentioned above: front and bottom. This parameter is also based on how the teacher would

respond to when they see one of their students engaged in an outside conversation or maybe even sleeping with their heads down.



Fig. 2.1: Shift in the head pose of students

2.4.2 Features of the proposed system

There is no need for the teacher to manually take attendance in the classroom because the system records a video and through further processing steps the face is being recognized and the attendance database is updated. This feature helps to automate the time consuming attendance process.

The other functionality of the system is to obtain the concentration level of the student. Once the facial features are detected then it determines the percentage of concentration level. The first criterion is to detect the motion of the student's head pose. This can be taken as a different feature to assess whether a student had interest in that particular topic or not.

The first two stages deal with capturing video feed as the input and segmenting it into frames for processing. The following stage is the processing module, which takes raw data as input and processes it to conclude decisions.

The last phase is the generation of application to showcase the raw data into a well visualized form which will be used thereby helping teachers to summarize the student's performance without indulging into the background working of the system.

2.4.3 Data collection

The data is collected by the camera which is arranged to be put in a suitable position. The position of the camera is such that all the students are visible and their faces are clear in the feed. Further, a call is made to the backend server which inputs the feed and does the processing work — identifying each student present in the class and marking their attendance, analysis of each present student's engagement level by the criterion given in the previous section.

In the proposed system, a raw video is being captured during a live classroom and is being provided to the scripts. The video stream is also used to compute the intellectual load using techniques that interconnect pupil extension and facial expressions to the intellectual load.

2.5 Objectives

On the basis of the knowledge about the existing systems, the objectives of the proposed systems are defined as follows:

- Mark the presence of each student in the classroom automatically during the lecture.
- Record each student's performance on the basis of their activeness in the class.
- Calculate the attention level of each student in the classroom which would help in generating an automated feedback for the teacher.

CHAPTER 3 - DESIGN AND IMPLEMENTATION

3.1 Design of the Proposed System

Raw video footage is collected from the camera which is installed in the classroom in such a way that all the students are visible in the camera. This footage will be used by the Application Programming Interfaces (APIs) in the back-end for analyzing the video to extract meaningful data. This meaningful data will finally get stored in the database that will be used by the front-end module. Scheme of the system proposed is as shown in Figure 3.1.

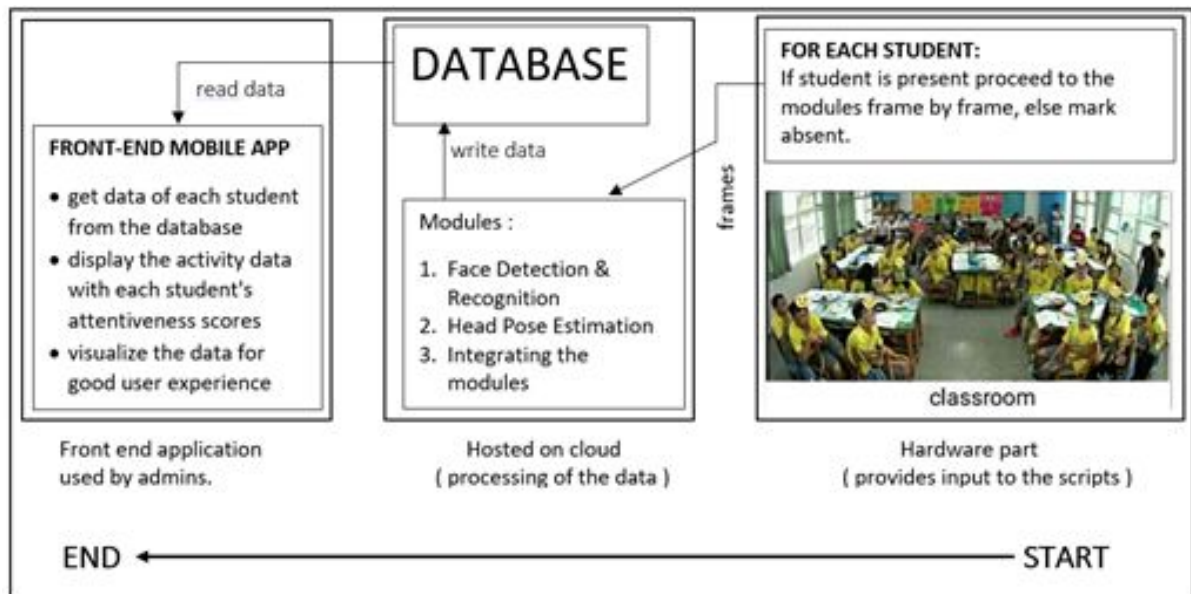


Fig. 3.1: Design of the proposed system

Detailed working of the APIs are as follows:

- Facial Detection/Recognition API:

This API helps to detect and recognize the faces of the students sitting in the class. It takes input as a frame from the video and checks for the total number of faces available on the frame. It uses two of the majorly known algorithms: the Haar Cascade Classifier Algorithm is for face detection and Local Binary Pattern Histogram algorithm for face recognition.

- Head Pose Estimation API:

This head pose estimation API calculates the angle of the head that is detected in front of the screen. It takes input from a frame in the video and checks for a single head that can be detected. Since the face of the person is a 3D object(world coordinates), it can rotate over all three axes - Yaw, Pitch and Roll. The 3D world coordinates will be translated to 3D camera coordinates.

- Integration API:

To integrate the modules an integration algorithm has been proposed. It integrates the face recognition and head pose estimation. It takes input from a frame consisting of a single face from a frame. It first recognizes the face of the student and then begins to calculate the angle of the head, based on which it categorizes the student's face into attentive or not.

3.2 IMPLEMENTATION

In the direction of achieving the objectives of the proposed system, it uses several tools and frameworks which reduce the code writing process by pre-defining frequently used functions. This section also describes the software that were used by the proposed system.

3.2.1 OpenCV

OpenCV (Open Source Computer Vision Library) is an “open-sourced library that includes many computer vision and machine learning algorithms”.

With the fact that each face is so much different and complicated, one simple test is not possible that can tell if it found the face or not. In contrast to this, there are thousands of small patterns and features that must be matched. The algorithms break the task of identifying the face into many smaller, bite-sized tasks, each of which is easy to solve.

As a computer vision library “OpenCV deals a lot with image pixels that are often encoded in a compact, 8- or 16-bit per channel, form and thus has a limited range of value”.

Furthermore, certain operations on images, like color space conversions, brightness/contrast adjustments, sharpening, complex interpolation can produce values that are not in the available range.

3.2.2 *NumPy*

- Numpy can be summarized as Numeric Python, a Python data analysis library that contains objects of various sizes and a set of objects for researching these structured objects.
- NumPy is Python's algebra line library, a very popular and widely used library because “most libraries available in PyData's environment rely on Numpy as one of their main building blocks”. In addition, it is quick and much more reliable. Numpy comes with two different flavors. These are: Vectors and Matrices. Here the vectors are 1D (one dimensional list), and the matrices are 2D (two dimensional) objects. It is noteworthy that matrices can also hold one row or column as well.
- Numpy allows developers to perform the following tasks:
 1. Design and fourier transformations
 2. Logical and mathematical operations
 3. Linear algebraic operations using structured functions

3.2.3 *Haar Cascade*

Haar Cascade is an “algorithm-based learning algorithm used to identify objects in a picture or video and is based on the concept of objects suggested by Paul Viola and Michael Jones”.

It is a machine learning technique “where Cascade's work is trained from many positive and negative images. After that it is used to find objects in other images.”

The algorithm has four stages:

1. Haar Feature Selection
2. Creating Integral Images
3. Adaboost Training
4. Cascading Classifiers

This method is used mainly for its ability to see faces and body parts through a photograph, on the contrary it can be trained to detect almost anything.

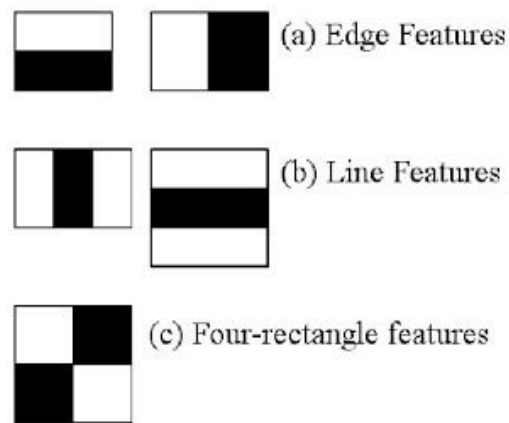


Fig. 3.2: Haar Cascade Features

Initially, the algorithm needs a lot of positive images of faces and negative images without faces to train the classifier. Then features are needed to be extracted.

First and foremost, we need to collect the Haar features. A Haar feature considers “adjacent rectangular regions at a specific location in a detection window, sums up the pixel intensities in each region and calculates the difference between these sums”. The three Haar Cascade features are shown in figure 3.2.

3.2.4 Local Binary Pattern Histogram

LBPH is a “machine learning classification algorithm used for facial recognition”.

This algorithm has 5 stages:

- Insert parameters: LBPH uses 4 parameters to say “radius, neighbor, grid x and grid y” where the radio is used to create a local binary pattern, neighbor number of sample models to create a spherical binary pattern, grid x denotes the number of cells in a straight, grid area y refers to the cell number in a vertical direction.in horizontal direction,grid y refers to number of cell in vertical direction.
- Training the Algorithm: For training the algorithm, dataset with facial images of the students were provided to the algorithm with a unique id of each student.
- Applying the LBH operation, as shown in figure 3.3:

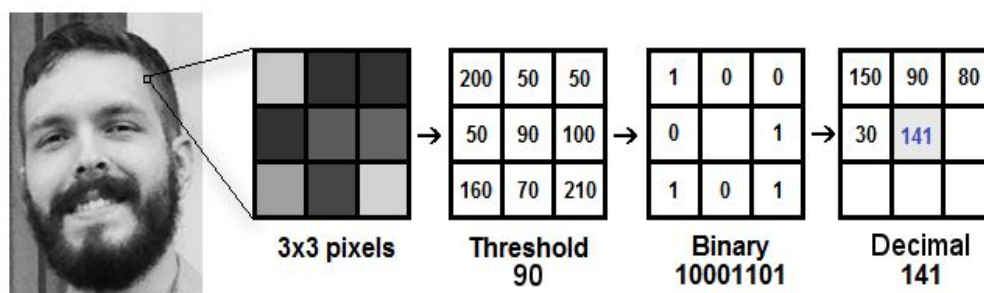


Fig. 3.3: LBPH operation on a single pixel.

The main aim of the LBPH operation is to “create an intermediate image that describes the original image in a more efficient manner”.

- Extracting the histogram :

The parameters of Grid X and Grid Y are used to separate the image generated from multiple grids. The process of histogram extraction is shown in figure 3.4.

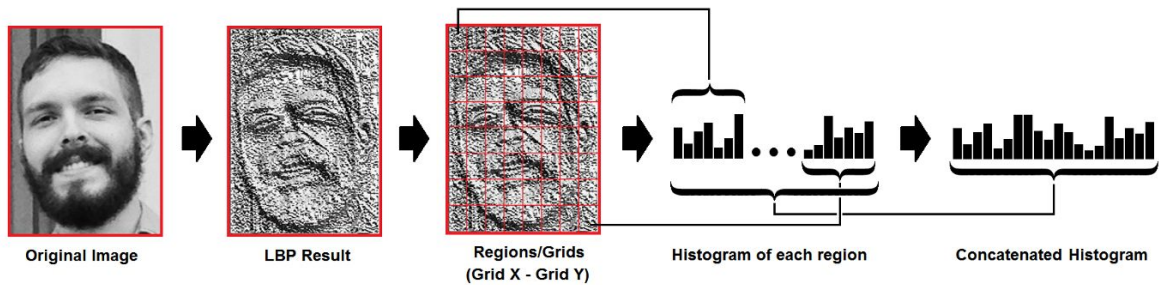


Fig. 3.4: Histogram extraction of the image.

- Performing Facial Recognition:

Each created histogram is used to represent each image from the training dataset. For obtaining an image matching the input image two histograms are compared and the image with the nearest histogram is returned.

3.2.5 *Head pose estimation*

In computer vision “the pose of an object refers to its relative orientation and position with respect to a camera”. The pose can be changed by either moving the object with respect to the camera, or the camera with respect to the object [7].

The pose estimation problem is often referred to as “Perspective-n-Point problem” or PNP in computer vision. The purpose is to determine the position of the object when the camera is calibrated, and the locations of n number of 3-D points on the object is known as well as the corresponding 2-D projections in the image.

A 3-D object has only two kinds of motions with respect to a camera, translation and Rotation. Moving the camera from its current 3-D location (‘X’, ‘Y’, ‘Z’) to a new 3-D location (‘X’, ‘Y’, ‘Z’) is called translation. It has 3 degrees of freedom — objects can move in the X, Y or Z direction. The camera can be rotated about the 3 axes, thus has three degrees of freedom.

There are many ways to represent the motion of rotation. It can be represented using Euler angles (roll, pitch and yaw), a direction of rotation (i.e. axis) and angle. The three Euler angles are depicted in Figure 3.5.

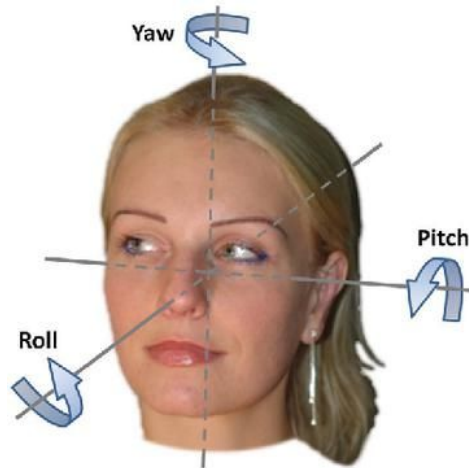


Fig. 3.5: Head pose estimation to find the angles - yaw, pitch and roll

So, estimating the pose of a 3D object means finding 6 numbers — three for translation and three for rotation.

To calculate the 3-D pose of an object in an image the following information is needed:

- 2-D coordinates of chosen points: 2-D (x,y) locations of a few points in the image is needed. Dlib's facial landmark detection provides 68 landmarks out of which 6 significant landmarks have been chosen by the proposed system: the tip of the nose, the chin, the left corner of the left eye, the right corner of the right eye, the left corner of the mouth, and the right corner of the mouth.
- Corresponding 3-D locations of the chosen points: The corresponding 3-D locations of the 2-D feature points is also needed. It is done in real-time by dlib.
- Intrinsic parameters of the camera : Generally, the focal length of the camera is needed to be known as well as the optical center in the image and the radial distortion parameters. So the camera is calibrated to find the approximate optical center by the

center of the image, the approximate focal length by the width of the image in pixels and it is assumed that radial distortion does not exist.

3.2.6. Integration API

The proposed integration API integrates face recognition and head pose estimation modules. It takes input from a frame from the video and then detects for a single face and recognizes it. After recognition, it calculates the angle of the head of the student. The following points describe the proposed algorithm in more detail:

1. The algorithm requires two input parameters: n and z where, n is the total number of frames in the video; z is the interval no. initialized by the user as desired.
2. Every time the algorithm runs, it captures each face present in the frame and crops them for better detailing.
3. The head angle function calculates the angle of the cropped frame containing the face.
4. The function returns a binary value depending upon the angles calculated.
5. The data is saved with the timestamp.
6. After the final iteration of the algorithm the data fetched is stored on MongoDB which is ready to be used by the app.
7. Face found more than 75% of iteration will be marked present for the class (83% for the case where $z=6$ and number of times face found=5).
8. The accuracy of the algorithm depends directly on z , i.e., the number of intervals.

3.2.7. Database

The proposed system uses MongoDB as its primary database. MongoDB is a NoSQL type database which uses JSON-like documents with schema. It is hosted on mLab cloud service platform. The database is remote, so it is connected using a URL. It contains useful data of every student's activity with a timestamp which is ready to be used by the mobile application. The schema of the data model is shown in figure 3.6.

QUERY RESULTS 1-1 OF 1

```
  _id: ObjectId("5e909699ed835073244c26f8")
  name: "student001"
  ✓ headActivity: Array
    ✓ 0: Object
      ts: 2020-04-10T15:53:56.271+00:00
      _id: ObjectId("5e909699ed835073244c26f9")
      angle: true
    ✓ presenceActivity: Array
      ✓ 0: Object
        ts: 2020-04-10T15:53:56.274+00:00
        _id: ObjectId("5e909699ed835073244c26fa")
        present: true
  > emotionActivity: Array
    __v: 0
```

Figure 3.6: Schema of the database

3.2.8. Mobile Application (Flutter)

The proposed system uses mobile application as the end product so that teachers could easily access the system and it will be feasible for them to use it anytime and anywhere. For this purpose Flutter is used to build mobile application. Flutter is a software development kit (SDK) which uses dart language to make an application for accessibility of the data in a proper and semantic manner.

Flutter is used because of the following reasons:

- It can develop applications for Android, IOS and most of all other operating systems.
- Famous for its best UI designs. It uses material design etc.
- Fast Response Time

- It is open source UI SDK created by Google

The merger of the data which was stored on MongoDB database was already processed and the mobile applications fetched the data directly by making a http get call.

The fetched data is modified in a user interactive state so that it becomes easy for teachers to interact and work along the app. A screenshot of the mobile application is shown in figure 3.7.

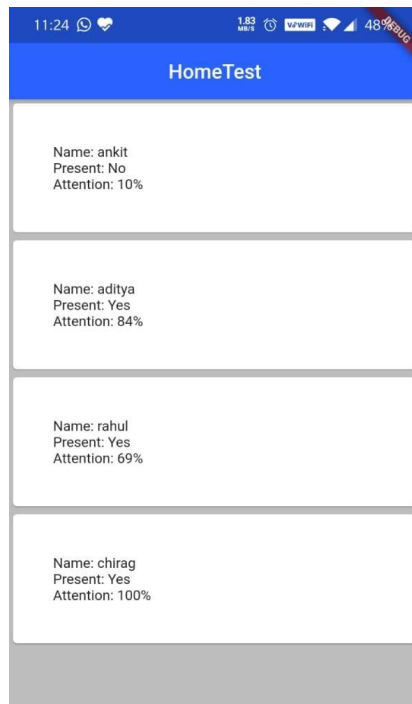


Fig 3.7: Fetched data into the app

CHAPTER 4 - RESULTS AND DISCUSSIONS

4.1 Results

Detection of faces is done by the proposed system which recognizes every face from the current frame. Fig 4.1 shows, the system recognizes the face of the student which was registered in the database and shows the name of the student associated with their face.



Fig. 4.1: Facial recognition of many students in classroom

If any student whose face is not present in the database and the student appears in front of the camera, then the system shows “unknown” associated with it as shown in Fig. 4.2.

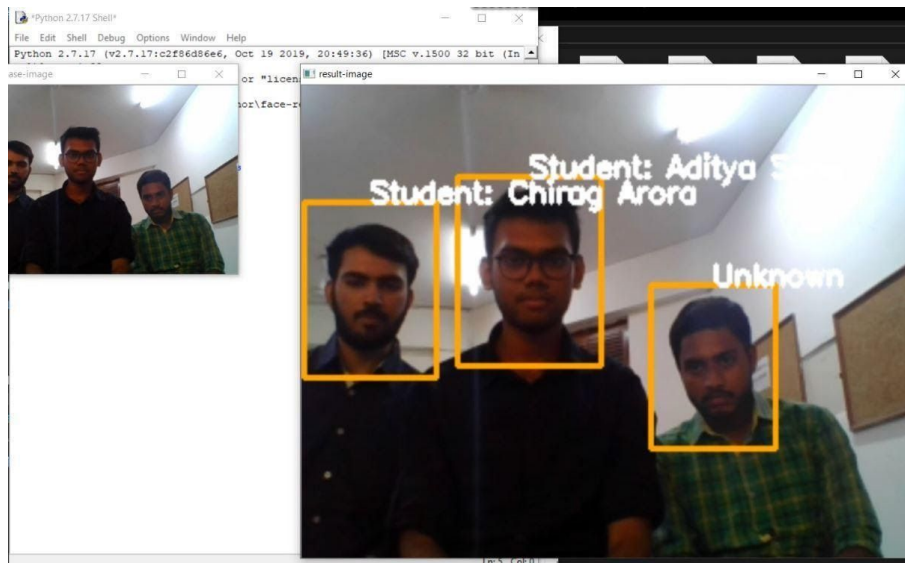


Fig. 4.2: Unknown person in frame

For better analysis of attentiveness of the students in the class head pose is being observed as shown in Fig. 4.3. This is done in a 3-dimensional model with all three axes of the model representing abscissa, ordinate and applicate respectively.

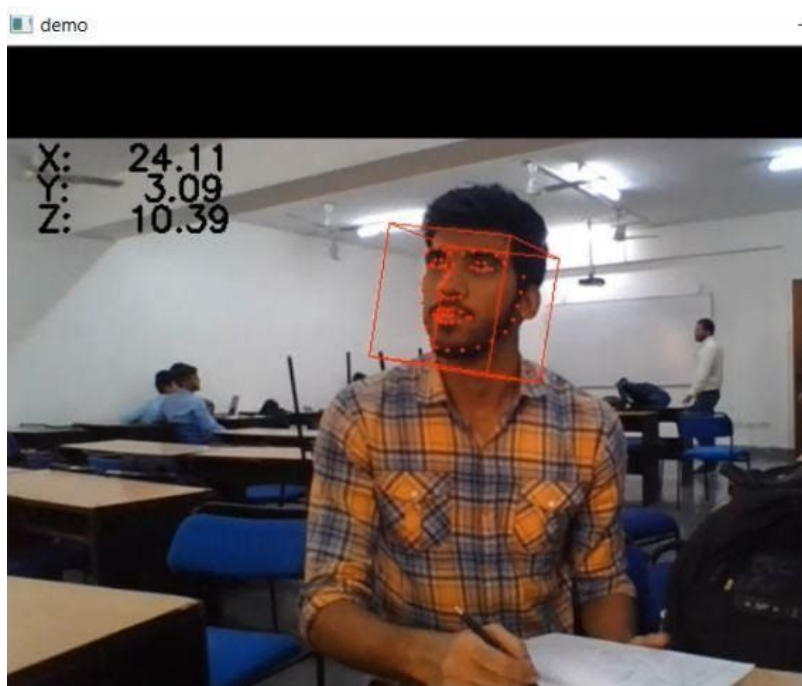


Fig. 4.3: Head pose estimation

4.2 Discussion

Proper feedback is given to the teacher about the student's behaviour and attention in the class. The teacher can see at any time any topic that the students are unable to understand from the lesson and can therefore improve their teaching style. This results in more efficiency and improved results of students in the classroom. This increases the interactivity level of the class which is very important for creating a healthy environment in a classroom.

Table 4.1 shows a comparative study between existing systems introduced by Zaletelj et al.[4], Whitehill et al.[5], Behnagh et al.[6] and the proposed system.

Table 4.1: Comparison between different existing systems and the proposed system

Parameters	Zaletelj et al.	Whitehill et al.	Behnagh et al.	Proposed System
Hardware Used	Kinect Sensor	Camera	Camera/Audio	HD Camera
Algorithm Used	KNN (K-Near Neighbour)	Support vector machine with Gabor feature (SVM(Gabor))	multi-modal sensing, behavior recognition algorithm	Local Binary Pattern Histograms (LBPH)
Features	Body, gaze and features of face	Face expressions	Facial expressions, body language	face detection and recognition, head pose estimation
Future Scope	Less practical because low-level data is obtained by Kinect sensor	The lack of correlation between engagement and learning	The types of data collected are not stretched to include self-reports of reading-related emotions	Accuracy can be increased

CHAPTER 5 - CONCLUSION AND FUTURE SCOPE

5.1 Conclusion

The proposed system's face recognizer works perfectly and can detect multiple faces at the same time. It can detect faces through a greater distance too and further improvements can be made by linking the webcam with our mobile phone thus increasing portability. This work is done based on various research papers, help and knowledge from teachers & friends, and motivation & support from the guide. The system uses many advanced techniques, such as haar cascading, Local Binary Pattern Histogram, numpy array manipulation, head pose estimation, which analyzes multimodal presentation movements and visual body language information and process information. Alternatively, the proposed system judges the class by assigning scores from different factors to determine the quality of the speech.

Irrespective of how versatile this project can be, we explored majorly on three domains in this project: first, automated examination of student's interest using head angle estimation, second, measure the attentiveness of students using their head positions, etc. and third, overcome the drawback of traditional attendance systems in classroom by determining the student's proportion of presence using facial recognition. In Real Time, the system analyzes the classroom using camera and running algorithms and eventually learns more about how class assesses a lecture. In the Back-end, this learned raw data is processed and designed for the front-end to give an easy UX and UI to give feedback to teachers and other staff members. The final goal is to improve the teaching and get the most out of students' behaviour and attention to provide them better engagement and interaction in the class.

5.2 Future Scope

There are some parameters that would likely be incorporated with the proposed system in the future:

1. Increase parameters of the system for measuring engagement level of a student and the classroom.
2. Improvement in performance by incorporating deep learning techniques to the project.
3. Hardware optimization is the most important aspect of this project as the camera needs to be a high resolution wide angle lens for covering more areas of the classroom while taking clear shots.
4. There is a scope in future to integrate emotion detection for better understanding of student attention.

REFERENCES

- [1] Larson, R. W., & Richards, M. H. (1991). Boredom in the middle school years: Blaming schools versus blaming students. *American journal of education*, 99(4), pp. 418-443.
- [2] Shernoff, D. J., Csikszentmihalyi, M., Schneider, B., & Shernoff, E. S. (2014). Student engagement in high school classrooms from the perspective of flow theory. In *Applications of flow in human development and education* pp. 475-494. Springer, Dordrecht. ISBN: 978-94-017-9094-9
- [3] Dunleavy, J., & Milton, P. (2009). What did you do in school today. Exploring the concept of student engagement and its implications for teaching and learning in Canada. Toronto: Canadian Education Association, 14(1), pp. 1-33.
- [4] Zaletelj J., & Košir A. (2017) “ Predicting students’ attention in the classroom from Kinect facial and body features” , *EURASIP Journal on Image and Video Processing*, 2017(1), pp. 80-91.
- [5] Whitehill, J., Serpell, Z., Lin, Y. C., Foster, A., & Movellan, J. R. (2014). The faces of engagement: Automatic recognition of student engagement from facial expressions. *IEEE Transactions on Affective Computing*, 5(1), 86-98.
- [6] Kim Y., Soyata T., & Behnagh R. F. (2018) “Towards emotionally aware AI smart classroom: Current issues and directions for engineering and education” *IEEE*, Vol. 6, pp. 5308-5331.
- [7] <https://www.learnopencv.com/head-pose-estimation-using-opencv-and-dlib/>

APPENDIX

1. hand_detection module

class HandGestureRecognition:

```
def __init__(self, cam_index=-1):
    self.cam_index = cam_index
    (self.opencv_ver, _, _) = cv2.__version__.split('.')
    self.frame = None
    self.frame_clone = None
    self.font = cv2.FONT_HERSHEY_SIMPLEX
    self.cam = cv2.VideoCapture(cam_index)
    self.face_hist = None
    self.face_rect = None
    self.hand_rect = None
    self.hand_hist = None
    self.hand_found = False
```

```
def find_hand(self):
```

```
    # Loop over the sliding window.
```

```
    hist_distances = []
```

```
    window_rectangles = []
```

```
    for window_rect in self.sliding_window(
```

```
        self.frame.shape,
```

```
        step_size=32,
```

```
        face_rect=(self.face_rect[0], self.face_rect[1], self.face_rect[2], self.face_rect[3])
```

```
    ):
```

```

window_hist = cv2.calcHist(
    [window_yrcrb],
    channels=[1],
    mask=None,
    histSize=[128],
    ranges=[0, 256]
)
d = cv2.compareHist(self.face_hist, window_hist, cv2.HISTCMP_CORREL)
if 0.6 < d < 0.9:
    hist_distances.append(d)
    window_rectangles.append(window_rect)
if hist_distances:
    max_d, max_d_idx = max((val, idx) for (idx, val) in enumerate(hist_distances))
    self.hand_rect = window_rectangles[max_d_idx]
    self.hand_found = True

# Calculate hand's histogram.
self.hand_hist = cv2.calcHist(
    [hand_image],
    channels=[0],
    mask=hand_mask,
    histSize=[128],
    ranges=[0, 256]
)
else:
    self.hand_found = False
    self.hand_rect = None
    self.hand_hist = None

```

```

def recognize_gesture(self):
    # Threshold the window to get the hand's binary mask.
    hand_image = self.frame_ycrnb[hand_y: hand_y + hand_h, hand_x: hand_x + hand_w,
1]
    hand_image = cv2.GaussianBlur(hand_image, ksize=(5, 5), sigmaY=1, sigmaX=1)
    hand_mask = cv2.morphologyEx(hand_image, cv2.MORPH_CLOSE, self.str_el)
    hand_mask = cv2.morphologyEx(hand_mask, cv2.MORPH_OPEN, self.str_el)
    hand_mask_in_frame = np.zeros(self.frame.shape[:2], np.uint8)
    hand_mask_in_frame[hand_y: hand_y + hand_h, hand_x: hand_x + hand_w] =
hand_mask

    thresh_deg = 80.0
    # Convexity hull based gesture recognition.
    contours = None
# Calculate the centroid to center the result's visualization and detect 0 gesture (closed fist).
    centroid = (hand_x + int(0.5 * hand_w), hand_y + int(0.6 * hand_h))
    valid_angles_count = 0
    possibly_zero = True
    gesture = min([5, valid_angles_count])
    if gesture <= 1 and possibly_zero:
        gesture = 0
    return hand_contour, defects, tuple(centroid), gesture

def sliding_window(self, frame_shape, step_size, face_rect):
    (face_x, face_y, face_w, face_h) = face_rect
    for y in range(face_y - int(face_h / 2), frame_shape[0] - int(1.5 * face_h), step_size):
        for x in range(face_x + int(2 * face_w), frame_shape[1] - face_w / 2, step_size):
            window_rect = [x, y, face_w, face_h]
            yield (window_rect)

```



```

def find_faces(self):
    # Face detection.
    if len(faces) == 1:
        for (x, y, w, h) in faces:
            x += w / 6
            y -= h / 10
            h += h / 10
            w -= w / 3
            self.face_hist = cv2.calcHist(
                [self.frame_ycrb],
                channels=[1],
                mask=face_mask,
                histSize=[128],
                ranges=[0, 256]
            )
    else:
        self.face_hist = None
        self.face_rect = None

def run(self):
    print('testing')
    key = 0
    # Press 'Esc' to quit.
    while key != 27:
        ret_val, self.frame = self.cam.read()
        if not ret_val:
            continue
        if not self.hand_found:

```

```

self.find_faces()
if self.face_hist is None or self.face_rect is None:
    continue

self.frame_clone = self.frame.copy()
cv2.putText(
    self.frame_clone,
    'Student detected: Aditya Sahu',
    (30, 30),
    self.font,
    1,
    (255, 255, 255),
    1
)
cv2.putText(
    self.frame_clone,
    'listening for activity..!',
    (30, 60),
    self.font,
    1,
    (255, 255, 255),
    1
)
cv2.rectangle(
    self.frame_clone,
    (self.face_rect[0], self.face_rect[1]),
    (self.face_rect[0] + self.face_rect[2], self.face_rect[1] + self.face_rect[3]),
    (0, 255, 0),
    2
)

```

```

    )

ret, self.hand_rect = cv2.meanShift(dst, tuple(self.hand_rect), self.mean_shift_term_crit)

# Refine the hand's mask.
hand_x, hand_y, hand_w, hand_h = self.hand_rect
hand_y = int(0.9 * hand_y)
self.hand_rect = [hand_x, hand_y, hand_w, hand_h]

# Gesture recognition.
hand_contour, defects, centroid, gesture = self.recognize_gesture()

# Visualization.
clone = self.frame.copy()
cv2.putText(clone, 'detecting emotion', (30, 30), self.font, 1, (255, 255, 255), 1)

self.cam.release()
cv2.destroyAllWindows()
exit()

if __name__ == '__main__':
    hand_gesture_recognition = HandGestureRecognition(cam_index=0)
    hand_gesture_recognition.run()

```

2. main module

```

def findStudentName(cap):
    old = funcDetectFace(cap)
    i=3

```

```

while i>0:
    print(funcDetectFace(cap))
    new = funcDetectFace(cap)
    if(old == new):
        i=i-1
    else:
        print('face lost/changed -> resetting')
        i=5
        old = new
    time.sleep(1)
return old

```

```

def findAngles(cap):
    #Find Angles for ctr seconds
    angles=[]
    i=5
    while i>0:
        ret=[]
        ret.append(funcDetectPose(cap))
        if ret == -1:
            print("no face found")
        elif ret == -2:
            print("camera not connected")
        ret.append(datetime.datetime.now())
        angles.append(ret)
        print(ret)
        i=i-1
        time.sleep(1)
    return angles

```

```

if __name__ == '__main__':
    ctr = 5
    cap = cv2.VideoCapture(0)
    studentName = findStudentName(cap)
    #studentName=("aditya")
    angles = findAngles(cap)
        #angles = [(-2.9541107103044504, -0.80001947719528999,
0.72367030867424731), (-6.1926197064548782, 3.3914647826761044,
-8.6323708765483271), (14.800405428511139, 4.9228644640433625,
6.8461287292403661), (20.707360555748618, 1.7134136849390902,
9.1795747334311315)]
    cv2.destroyAllWindows()
    cap.release()
    row = [studentName]
    for i in angles:
        row.append(i)
    print(row)

```