COMPUTER VISION-BASED WARNING SYSTEM OF LEARNERS ENGAGEMENT SHIFT IN ONLINE LEARNING PLATFORMS

A PROJECT REPORT

Submitted by

OM RAJPUT [RA1911032010028]

SUYASH NIGAM [RA1911032010027]

Under the guidance of

Dr. TYJ NAGA MALLESWARI

(Associate Professor, Department of Networking and Communications)

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

with specialization in INTERNET OF THINGS

SRM INSTITUTE OF SCIENCE & TECHNOLOGY

Deemed to be University u/s 3 of UGC Act, 1956

DEPARTMENT OF NETWORKING AND COMMUNICATIONS

COLLEGE OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR- 603 203

MAY 2023

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR - 603203

BONAFIDE CERTIFICATE

Certified that major project report titled "**COMPUTER VISION BASED WARNING SYSTEM OF LEARNERS ENGAGEMENT SHIFT IN ONLINE LEARNING PLATFORMS**" is the bonafide work of "**OM RAJPUT [RA1911032010028], SUYASH NIGAM [RA1911032010027]"** who carried out the major project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE	**SIGNATURE**
Dr. TYJ Naga Malleswari	Dr. Annapurani K
SUPERVISOR	**HEAD OF THE DEPARTMENT**
Associate Professor	Professor
Dept. of Networking and Communications	Dept. of Networking and Communications

Signature of the Internal Examiner **Signature of the External Examiner**

Department of Networking and Communications

SRM Institute of Science & Technology

Own Work* Declaration Form

Degree/ Course : B. Tech CSE with specialization in Internet of Things

Student Name : Om Rajput

Registration Number : RA1911032010028

Student Name : Suyash Nigam

Registration Number : RA1911032010027

Title of Work : Computer Vision Based warning systems for learners engagement shift in Online Learning Platforms

We hereby certify that this assessment compiles with the University's Rules and Regulations relating to Academic misconduct and plagiarism**, as listed in the University Website, Regulations, and the Education Committee guidelines.

We confirm that all the work contained in this assessment is our own except where indicated, and that We have met the following conditions:

- Clearly references / listed all sources as appropriate.
- Referenced and put in inverted commas all quoted text (from books, web, etc)
- Given the sources of all pictures, data etc. that are not my own.
- Not made any use of the report(s) or essay(s) of any other student(s) either past or present
- Acknowledged in appropriate places any help that we have received from others (e.g., fellow students, technicians, statisticians, external sources)
- Compiled with any other plagiarism criteria specified in the Course handbook / University website.
- We understand that any false claim for this work will be penalized in accordance with the University policies and regulations.

DECLARATION:
We are aware of and understand the University's policy on academic misconduct and plagiarism and we certify that this assessment is our own work, except where indicated by referring, and that we have followed the good academic practices noted above.
Om Rajput(RA1911032010028) : Suyash Nigam(RA1911032010027) :

ACKNOWLEDGEMENTS

We express our humble gratitude to **Dr C. Muthamizhchelvan**, Vice-Chancellor, SRM Institute of Science and Technology, for the facilities extended for the project work and his continued support. We extend our sincere thanks to Dean-CET, SRM Institute of Science and Technology, **Dr T.V.Gopal**, for his valuable support.

We wish to thank **Dr Revathi Venkataraman**, Professor & Chairperson, School of Computing, SRM Institute of Science and Technology, for her support throughout the project work. We are incredibly grateful to our Head of the Department, **Dr. Annapurani**.**K** Professor, Department of Networking and Communications, SRM Institute of Science and Technology, for her suggestions and encouragement at all the stages of the project work.

We want to convey our thanks to our program coordinator **Dr. A Suresh**, Associate Professor, and Panel Head, **Dr. C Malathy** Professor, Department of Networking and Communications, SRM Institute of Science and Technology, for their inputs during the project reviews and support. We register our immeasurable thanks to our Faculty Advisor, **Dr. P Vigneshwaran**, Associate Professor, Department of Networking and Communications, SRM Institute of Science and Technology, for leading and helping us to complete our course.

Our inexpressible respect and thanks to my guide, **Dr. TYJ Naga Malleswari**, Associate Professor, Department of Networking and Communications SRM Institute of Science and Technology, for providing me with an opportunity to pursue my project under his/her/their mentorship. She provided me with the freedom and support to explore the research topics of my interest. Her passion for solving problems and making a difference in the world has always been inspiring.

We sincerely thank the Networking and Communications Department staff and students, SRM Institute of Science and Technology, for their help during our project. Finally, we would like to thank parents, family members, and friends for their unconditional love, constant support, and encouragement.

Om Rajput

Suyash Nigam

ABSTRACT

Online students participate in a variety of educational pursuits, including reading, writing, viewing instructional videos, taking online exams, and participating in online meetings. While taking part in these educational activities, they exhibit a wide range of levels of interest, including boredom, aggravation, delight, indifferent, confusion, and learning gain. In order to provide the online students with individualized pedagogical support through interventions, online educators need to precisely and efficiently identify the level of engagement their online students currently possess. In the context of online education, this study gives an overview of the most recent developments in the field of engagement detection.

The technique of determining whether or not individuals are paying attention is referred to as the measuring engagement. The term "student engagement" refers to the degree to which students mentally and emotionally involve themselves in their academic work. In order to assist in the process of improving the figures, defining a clear procedure to assess and understand the patterns in engagement measurement can go a long way toward contributing to the improvement.

Throughout the course of this project, we have had the opportunity to review the prior research that was carried out by a variety of academics. As a result of the work that we did, we now have the knowledge necessary to develop our very own deep learning model for engagement detection. In addition to that, this model is supported with a web application that simulates the operation of a platform for holding online meetings. The goal of this project is to accomplish the integration of the aforementioned model into the application so that real-time data can be obtained from online meetings

The identification of emotions served as the foundation for our first set of models, and we intended to use them to develop a metric for gauging levels of user engagement. This strategy was quickly elevated to a higher level when an original engagement detection convolutional neural network was conceived of as a potential improvement. The emotion detection model will continue to play an important part for the fact that we are now in a position to make a decision that is better informed as a result of having more data.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.

CHAPTER NO.	TITLE	PAGE NO.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

CNN	Convolutional Neural Network
ML	Machine Learning
DL	Deep Learning
NN	Neural Network
App	Application
RNN	Recurrent Neural Network
ED	Emotion Detection
ReLU	Rectified Linear Units
UI	User Interface
CSS	Cascading Style Sheets
HTML	Hypertext Markup Language
WebRTC	Web Real-Time Communication
UML	Unified Modeling Language
SVM	Support Vector Machine
KNN	K Nearest Neighbors

CHAPTER 1

INTRODUCTION

With the rise of technology, virtual meetings have become a popular method of communication for people in different locations. Virtual meetings use video conferencing technology to allow participants to communicate and collaborate remotely in real-time. This has become especially important in the wake of the COVID-19 pandemic, as many people have had to work and communicate from home. While virtual meetings may seem very different from traditional in-person conferences, they share a common goal: bringing people together in a special meeting room. In an in-person conference, everyone is physically present in the same room, whereas in a virtual meeting, participants can be located anywhere in the world as long as they have an internet connection.

One of the primary advantages of virtual meetings is that they enable more flexible and efficient communication. Participants can participate in the meeting without having to travel. This saves time and money while also minimizing travel's environmental impact. Furthermore, because participants do not need to factor in travel time, virtual meetings can be planned more easily. Virtual meetings can also be recorded, allowing participants to re-watch the meeting or share it with others who were unable to attend. This is especially beneficial for training sessions or presentations when participants may want to review the material afterwards.

1.1 Computer Vision

Computer vision is a subfield of artificial intelligence that tries to teach machines how to perceive and comprehend visual data in the same way that people do. It entails the creation of algorithms and procedures that allow computers to analyze, process, and comprehend images and videos.

In recent years, computer vision has made tremendous advancements due to the availability of large datasets and powerful machine learning algorithms. Some of the practical applications of computer vision include object recognition, face detection, image and video analysis, augmented reality, and autonomous navigation.

Object recognition is one of the most important applications of computer vision. It involves training algorithms to identify and classify objects in images and videos accurately. Object recognition is used in a wide range of applications, from identifying cancer cells to self-driving cars.

Another important application of computer vision is face detection, which involves identifying human faces in images and videos. This technology has been used for security purposes, such as in surveillance systems, and for improving the user experience in mobile devices.

Image and video analysis is another area where computer vision has shown significant progress. This technology enables computers to extract valuable information from images and videos, such as identifying patterns, detecting anomalies, and tracking objects over time.

Augmented reality is an exciting application of computer vision that overlays digital information on the real world, creating an immersive experience for the user. This technology has been used in gaming, advertising, and education, among others. Finally, autonomous navigation is an application of computer vision that enables machines to navigate the world without human intervention. This technology has been used in self-driving cars, drones, and robots, among others, enabling them to safely and efficiently navigate their surroundings.

In conclusion, computer vision is a rapidly growing field that has numerous practical applications. With the increasing availability of visual data and the development of powerful machine learning algorithms, we can expect to see many more exciting applications of computer vision in the years to come.

1.2 Types of Computer Vision Techniques

It is a vast field that involves the development of algorithms and techniques to enable machines to interpret visual data. Different types of computer vision are used to solve specific problems, and they have various applications. Here are some of the main types of computer vision:

- Image classification: This type of computer vision involves training machines to categorize images into different classes or categories. For example, a machine can be trained to identify an image as a cat, a dog, or a car. Image classification is used in various applications, including image search and content filtering.
- Object detection: Identifying and detecting items inside an image or video is a sort of computer vision. Object detection is utilized in a variety of applications, including autonomous vehicles, surveillance, and robotics.
- Semantic segmentation: This type of computer vision involves segmenting an image into different regions and assigning a label to each segment. Semantic segmentation is used in applications such as medical image analysis and autonomous navigation.
- Instance segmentation: This type of computer vision involves identifying and delineating individual objects within an image or video. Instance segmentation is used in applications such as autonomous driving and robotics.
- Optical character recognition (OCR): This type of computer vision involves recognizing and interpreting text within an image or video. OCR is commonly used in applications such as document scanning and automated data entry.
- Face recognition: This type of computer vision involves identifying and verifying the identity of a person based on their facial features. Face recognition is used in applications such as security systems, mobile devices, and social media.
- Motion analysis: This type of computer vision involves analyzing and interpreting the motion of objects within an image or video. Motion analysis is used in applications such as sports analytics, surveillance, and robotics.

1.3 Introduction to Object Detection

Computer vision is the process of identifying and detecting objects inside an image or video. Object detection is used in many different applications, such as autonomous cars, surveillance, and robots.

There are two main approaches to object detection: the traditional computer vision approach and the deep learning approach. The traditional approach involves extracting features from the image using handcrafted algorithms such as Histogram of Oriented Gradients (HOG) and Local Binary Patterns (LBP) and then using a classifier, such as a Support Vector Machine (SVM), to identify the object. However, this approach has limitations in terms of accuracy and scalability.

In contrast, the deep learning approach to object detection has shown significant progress in recent years. It involves training a convolutional neural network (CNN) to learn features directly from the image and use them to detect objects. One of the most popular deep learning-based object detection frameworks is the region-based convolutional neural network (R-CNN) family of algorithms, which includes Faster R-CNN, Mask R-CNN, and Cascade R-CNN. These frameworks use a combination of CNNs and region proposal algorithms, such as selective search or region proposal network (RPN), to generate region proposals and classify them into different object classes.

Object detection has numerous applications in various fields, such as robotics, autonomous driving, and surveillance. For example, in autonomous driving, object detection is used to identify and locate pedestrians, vehicles, and other objects to avoid collisions. In robotics, object detection is used to detect and locate objects to enable robots to perform tasks such as object manipulation and assembly. In surveillance, object detection is used to detect and track people and objects of interest.

1.4 Existing Systems

Student engagement is a critical factor in determining student success in online learning environments. As online learning continues to grow in popularity, there is an increasing need to develop accurate and reliable methods for measuring student engagement. The purpose of this literature review is to provide a comprehensive overview of the existing literature on engagement detection in online learning, including the methods used to measure engagement and the relationship between engagement and student outcomes.

1.4.1 Measurement of Engagement

Engagement in online learning is a complex construct that can be measured using a variety of methods. Behavioral measures, such as the amount of time spent on a task or the number of interactions with the learning environment, have been widely used to measure engagement. These measures are easy to collect and provide valuable insights into students' behaviors and activities in the learning environment.

In addition to behavioral measures, physiological measures have been used to detect engagement in online learning. These measures include electroencephalography (EEG), electrodermal activity (EDA), and heart rate variability (HRV). These measures provide a more objective and accurate assessment of engagement by detecting changes in physiological responses associated with engagement.

1.4.2 Relationship between Engagement and Outcomes

Numerous studies have investigated the relationship between engagement and student outcomes in online learning. Overall, the findings suggest that higher levels of engagement are associated with better learning outcomes, such as higher grades and greater satisfaction with the learning experience.

One study by Kreijns et al. (2013) found a positive relationship between engagement and academic performance in online learning. The study used a self-report measure of engagement and found that students who reported higher levels of engagement had higher grades and were more satisfied with the learning experience.

Similarly, a study by Huang et al. (2016) used a physiological measure of engagement (EEG) and found that students who showed greater engagement during a learning task performed better on a subsequent test of knowledge retention. The findings suggest that physiological measures of engagement can provide valuable insights into the relationship between engagement and learning outcomes.

1.4.3 Machine Learning algorithms for Engagement Detection

Recent studies have explored the use of machine learning algorithms for analyzing engagement data in online learning environments. These algorithms can analyze large amounts of data and identify patterns that are not easily detectable using traditional methods.

For example, a study by de Freitas et al. (2017) used machine learning algorithms to analyze data from a learning analytics platform and identify patterns of engagement associated with higher levels of academic achievement. The study found that students who demonstrated higher levels of engagement early in the course were more likely to achieve higher grades.

Another study by Liao et al. (2018) used machine learning algorithms to analyze data from an online learning platform and predict student dropout rates. The study found that engagement measures, such as the number of logins and the amount of time spent on the platform, were significant predictors of student dropout.

1.4.4 Limitation and Future Directions

Behavioral measures may not accurately capture the complex nature of engagement, and physiological measures can be invasive and difficult to collect. In addition, machine learning algorithms are only as accurate as the data they are trained on, and there is a need for more high-quality engagement data to improve the accuracy of these algorithms.

Future research in this area should focus on developing more sophisticated methods for measuring engagement, including the use of emerging technologies such as wearable sensors. Additionally, there is a need for more research on the relationship between engagement and specific learning outcomes, such as knowledge retention and skill acquisition, to better understand the mechanisms underlying the relationship between engagement and learning success.

1.5 Proposed Systems

As the world navigates the post challenges posed by the COVID-19 pandemic, the education sector has been forced to adapt rapidly. Online learning has become the new norm, but it poses new challenges, particularly when it comes to measuring student engagement. Measuring student engagement is important because it provides instructors with insights into how students are participating in the class and where they might need additional support.

Our project focuses on developing a custom CNN architecture that can detect whether a student is engaged or not in an online class. We plan to integrate this model into a web application that simulates the experience of holding an online meeting with real-time data analysis. The application will mimic the functionality of a teacher in a real-world classroom, while also incorporating new and creative features. To achieve this, we will use WebRTC, a JavaScript library that enables real-time communication.

In the backend, we will use Deep Learning and ReLU, which provides non-linearity to the model. Our goal is to develop a neural network that is completely connected, with the softmax activation function implemented in the last connected layer, to produce a probability between 0 and 1 for each class the model is predicting.

Our first set of models will focus on identifying emotions, which will serve as a foundation for developing a metric to gauge user engagement. This approach is based on the assumption that engaged students tend to display certain facial expressions, such as smiling or nodding their heads, while disengaged students might appear distracted or bored.

As we progress, we plan to improve our model by developing an original engagement detection CNN. This model will go beyond just identifying facial expressions and will take into account other factors that might indicate engagement or disengagement. For example, we might consider the amount of time a student spends looking at the screen, the frequency of their interactions, or the pace at which they complete tasks.

Ultimately, our aim is to accurately measure student attentiveness in an online classroom setting. However, this is easier said than done. One of the challenges we face is developing a reliable and accurate metric for measuring engagement. While facial expressions can provide some indication of engagement, they are not always reliable, and different students might exhibit different expressions for the same level of engagement.

Another challenge we face is the issue of privacy. We will need to collect data on students' facial expressions and other behaviors, which raises concerns about privacy and data protection. We plan to address this issue by ensuring that all data collected is anonymized and by obtaining consent from students and their parents/guardians.

Furthermore, the performance of the model might vary depending on the context in which it is used. For example, the model might work well in a classroom setting but might struggle in a one-on-one tutoring session. We plan to address this challenge by conducting extensive testing and by fine-tuning the model based on the feedback we receive.

In conclusion, our project aims to develop a custom CNN architecture that can accurately measure student engagement in an online classroom setting. While we anticipate facing several challenges, we believe that this project has the potential to significantly enhance the online learning experience for students and instructors alike. By providing real-time insights into student engagement, our model can help instructors identify areas where students might need additional support, ultimately leading to improved academic outcomes.

1.6 Objectives

1. Develop a custom CNN architecture: The first objective is to develop a convolutional neural network (CNN) that is specifically designed for measuring student engagement in an online classroom setting. This involves selecting appropriate layers, filters, and activation functions, and fine-tuning the model through a process of trial and error.
2. Integrate the model into a web application: The second objective is to integrate the CNN model into a web application that simulates the experience of holding an online meeting with real-time data analysis. This involves using WebRTC, a JavaScript library that enables real-time communication, to stream video and audio data from students' webcams and microphones, and to transmit this data to the backend for analysis.
3. Develop a reliable and accurate metric for measuring engagement: The third objective is to develop a reliable and accurate metric for measuring student engagement based on the data collected by the model. This involves selecting appropriate features to analyze, such as facial expressions, gaze direction, and interaction frequency, and developing an algorithm for combining these features into a single metric. The metric should be sensitive enough to detect subtle changes in engagement, while also being robust to noise and individual differences.

1.7 Innovation

The innovation in the above article is the development of a custom convolutional neural network (CNN) architecture that can accurately measure student engagement in an online classroom setting. The use of a CNN is a powerful and effective approach for processing visual data such as facial expressions, which can provide insights into a student's level of engagement. The use of a custom architecture allows for the optimization of the model's performance for the specific task of measuring student engagement.

The approach of using facial expressions and other behavioral factors as a means of gauging student engagement is a novel and potentially more accurate method compared to traditional methods such as self-reported surveys. This is because facial expressions and other behaviors provide real-time, objective indicators of a student's engagement level, as opposed to subjective self-reports that may be influenced by a range of factors.

A novel and innovative approach to the problem of gauging student interest in an online learning environment is the incorporation of the engagement detection CNN into a web application that mimics the experience of attending an online conference with real-time data analysis. The smooth integration of the engagement detection CNN into the web application and the availability of real-time feedback on student participation during online lessons are made possible by the use of WebRTC, a JavaScript package that provides real-time communication.

The use of deep learning techniques and the rectified linear unit (ReLU) activation function in the backend of the model provides non-linearity and allows for the development of a completely connected neural network architecture. The use of the softmax activation function in the last connected layer of the model produces a probability between 0 and 1 for each class the model is predicting, allowing for accurate and reliable predictions of student engagement levels.

However, the development of this innovative solution is not without challenges. One of the main challenges is developing a reliable and accurate metric for measuring engagement. While facial expressions and other behaviors can provide some indication of engagement, they are not always reliable, and different students might exhibit different expressions for the same level of engagement. To address this challenge, the model will need to be trained on a large and diverse dataset to ensure that it can accurately identify a range of expressions and behaviors.

Another challenge is the issue of privacy, as the model will need to collect data on students' facial expressions and other behaviors, raising concerns about privacy and data protection. To address this issue, all data collected will be anonymized, and consent will be obtained from students and their parents/guardians.

1.8 Hardware and Software Requirements

Hardware requirements refer to the necessary physical components apart from the source device needed to run a system, such as a computer or server, while software requirements refer to the necessary programs and applications needed to run the system, such as an operating system or programming language. Both hardware and software requirements must be carefully considered and met to ensure the system operates effectively and efficiently.

1.8.1 Hardware Requirements:

- Processor: Intel i5 or above
- RAM: 4GB
- Harddisk: 10GB
- Camera minimum 480P resolution
- Stable Internet Connection

1.8.2 Software Requirements:

- Operating system: Windows 7 or above
- Python, JavaScript, React.Js
- Suitable code editor such as Microsoft visual studio code
- Suitable server like Firebase

CHAPTER 2

LITERATURE SURVEY

A literature survey, also known as a literature review, is a critical and comprehensive evaluation of existing literature related to a specific research topic or question. It involves an extensive search and review of relevant literature, including books, articles, and other scholarly sources, to provide an overview of the current state of knowledge in the field. The purpose of a literature survey in a report is to provide a detailed and comprehensive analysis of the existing literature related to the research question or topic. This analysis helps the reader to understand the context of the research and the current state of knowledge in the field. It also helps the researcher to identify gaps in the existing literature and develop research questions and hypotheses based on the gaps.

2.1 Related Works

Zeyad Abdulhameed and colleagues[1] applied convolutional neural networks (CNNs) to classify student eye-gaze participation in educational environments. They utilized pre-trained models based on the MobileNet architecture, which is a popular deep learning model for image classification tasks, initially trained on the ImageNet dataset. They employed the softmax activation function to estimate student engagement or disengagement based on the output of the second layer of the CNN. During training, the Root Mean Square Propagation (RMS prop) optimization algorithm was used to reduce the error of the model. In addition, the researchers employed regularization techniques such as dropout and batch normalization to prevent overfitting of the model to the training data. The Figure 2.1 figuratively represents the process of feature extraction using MobileNet Model architecture[1].

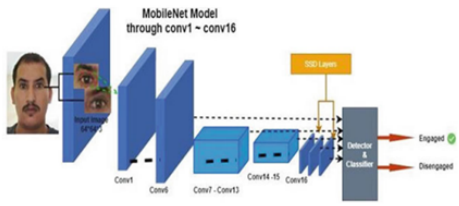

Figure 2.1. MobileNet model architecture

The CNN design, according to Chakradhar & Kumar et al. [2], has several layers like convolution-2D, max-pooling-2D, flatten, hidden, and softmax layers, like shown in Figure 2.2.

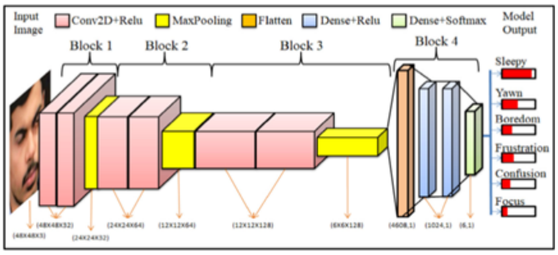

Figure 2.2.Model architecture of CNN

As a pre-trained face recognition model, a multi-task cascade convolution neural network (MTCNN) was utilized. Any non-frontal faces from either the input image, including left- or right-skewed, upward- or downward-facing faces, are removed by the head position identification procedure. Y. -Y. Ou et al[3] used SFPs to observe human expression. Identity is confirmed by body type and facial skeleton. Local Binary Pattern Histogram measures facial light and shadow (LBPH). The skeleton dictates body type.Emotion recognition uses facial landmarks and HC-SVM. SFPs are used to monitor facial action units during emotion recognition. SFPs use API facial landmarks (API). Face texture feature extraction and identity recognition use LBPH and KNN. Joint count determines body type. Figure 2.3 represents the architecture of Support Vector Machine (also pronounced as SVM) for action unit recognition with hierarchical-coherence.

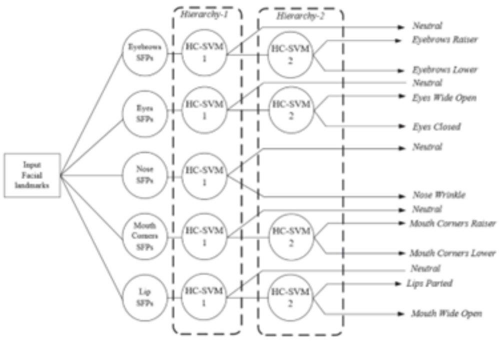

Figure 2.3. The architecture of Hierarchical-Coherence Support Vector Machine for action unit recognition.

In their study, S. Deniz et al[4] proposed the use of OpenCV, a widely used facial recognition library, as the main dependency for their program. Haar and Local Binary Pattern Histograms (LBPH) algorithms are implemented in OpenCV for face detection and recognition. The Haar algorithm works by evaluating the highlights and shadows of the grayscale images using Haar features. This allows the algorithm to effectively identify facial regions in the image. On the other hand, the LBPH algorithm is used for feature extraction by studying texture and structure in tiny pixel regions. EigenFaces and FisherFaces algorithms were used with LBPH for facial recognition.

The program is capable of detecting a range of emotions including fury, disdain, disgust, fear, happiness, indifferent, melancholy, and astonishment. However, the application only monitors happiness, sadness, wrath, and neutrality despite detecting eight emotions. The Figure 2.4 represents the input and output of each level of the program module. The use of OpenCV with Haar and LBPH algorithms for facial recognition allows for accurate and efficient detection and recognition of facial features, making it a popular choice for many facial recognition applications.

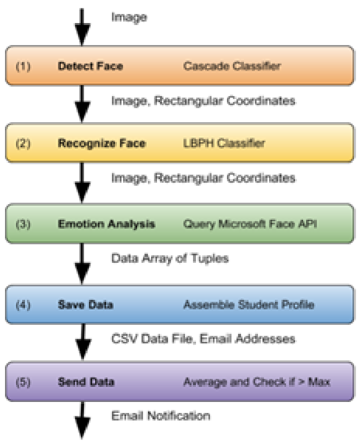

Figure 2.4. Representation of program module

P. Chiranjeevi et al[5] suggested that each input shape is aligned using Procrustes analysis. To accommodate for CLM alignment difficulties caused by posture changes, a statistical texture model for each KEY point and use structural similarity to identify change. This section outlines our algorithm-Procrustes analysis

● Emotional highlights:

● KEYPOINT tracking

● Patch Representation

● Multi-neighbor structural comparison

● Fusion of KE Point Pair Distances

● CLM to track N facial feature points in each frame.

After detecting changes in status in each location (cheek, brow, and mouth), the proposed approach employs just the relevant AUs for emotion categorization, as opposed to existing ER systems, which give characteristics to all AUs[5]. Figure 2.5 depicts an illustrated diagram of the pre-processor.

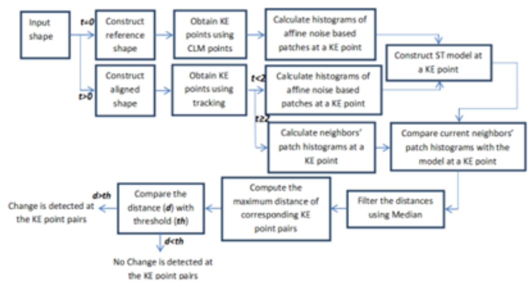

Figure 2.5. Illustrative diagram of the pre-processor.

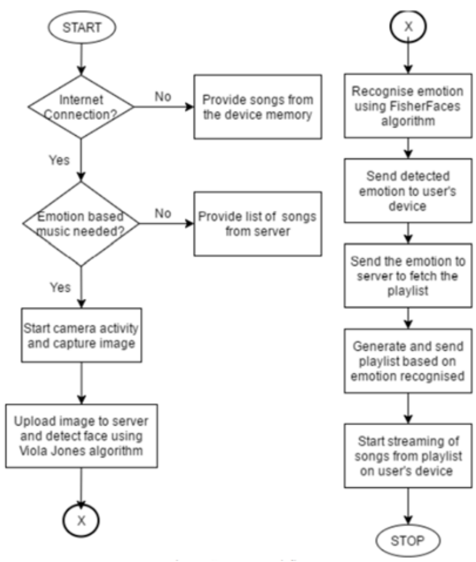

Figure 2.6 . System workflow

A. V. Iyer et al[6] in their study have used the following techniques for Face detection:

- Canny Edge Identification
- Viola Jones

Voice, face, and body language can all convey emotions. This study examines face expressions as identifiers, like shown in Figure 2.6.

Apicella et al. [7] describe an EEG-based technique to identify emotional and cognitive involvement during a learning task. Two Common Spatial Pattern (CSP) algorithms that have each been trained to assess Cognitive and Emotional Engagement provide a set of features to two Support Vector Machine (SVM) classifiers as shown. A baseline removal followed by a TCA procedure is offered during the classifier's training phase only for cross-subject data. The interpretation of the engagement assessment is shown metaphorically in Figure 2.7 below.

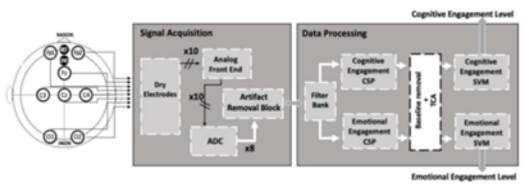

Figure 2.7. Student Engagement architecture

Mustafa and Zdemir et al. have proposed a methodology that combines several image processing techniques for the recognition of images and the estimation of head posture to measure student engagement in real-time. This methodology involves the use of the SECS (Spatial Enhanced Correlation-based Subspace) algorithm for image recognition, which is capable of detecting features of interest in an image, and the plot of directional gradients for face recognition. Additionally, the PnP (Perspective-n-Point) algorithm is employed for estimating the head posture of the students, which helps to infer their level of engagement.

The proposed methodology also uses SVM (Support Vector Machines) for classification, which is a machine learning algorithm that can be used to classify data into different categories like shown in the Figure 2.8. The SVM algorithm is used to segment the screenshots of the students' faces into two categories: "Engaged" or "Not Engaged". The SVM algorithm achieves this segmentation outcome by using the features extracted from the SECS algorithm, plot of directional gradients, and head posture estimation from the PnP algorithm.

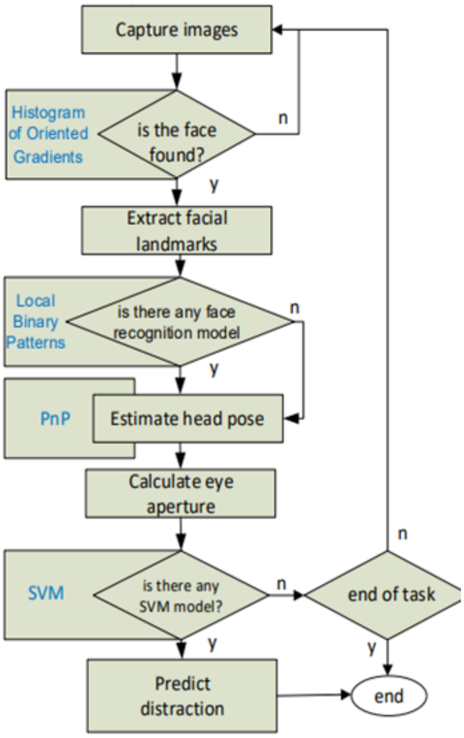

Figure 2.8. Flowchart of main module of SECS

The method used by Ayouni S et al [9] was based on the student's Grade Point Average (GPA) and the instructor's evaluation. It includes the category target variable (class), which uses GPA to quantify student participation as shown in the graph below named Figure 2.9. Classification prediction methods that are employed include DT or Decision Tree classification, SVM or Support Vector Machine, and ANN or Artificial Neural Network[9].

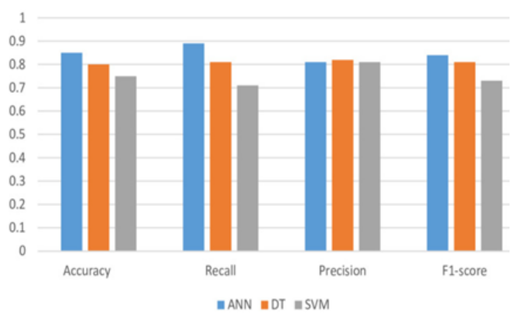

Figure 2.9. Performance metrics of ANN, DT and SVM.

Ali Abedi et al[10] utilized a 3D CNN, spatiotemporal characteristics are retrieved from videos. The retrieved features from 3D CNN are passed to ResNet, whose output is then passed to TCN for temporal analysis. In our design, ResNet is used to extract spatial information from frames, and the outputs of ResNet are passed to TCN for temporal analysis as shown in the below image labeled Figure 2.10.

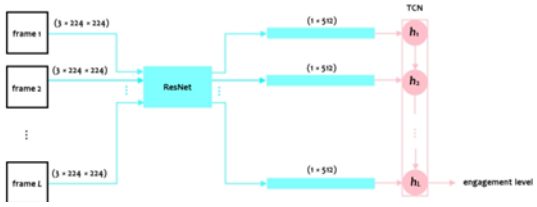

Figure 2.10. The end to end architecture, addition of ResNet and TCN, for engagement level detection

So, for our study, we decided to build a custom CNN architecture to determine whether or not students are engaged in a class.The model that we came up with is highly rudimentary and decides whether a student is engaged or not based on only the facial expressions.

The growth of virtualization technology has led to an increase in the number of VMs and data centers, resulting in duplication of data for various purposes[11]. Data Deduplication is a technique that detects and eliminates duplicate data, reducing the consumption of resources like storage space and network bandwidth[11].

Cloud computing enables resource sharing through virtualization, which optimizes computer system utilization[12]. Virtual machine migration addresses load balancing and power consumption issues in the cloud environment[12].

Virtual machine migration offers advantages such as fault tolerance and load balancing[13]. The pre-copy approach is often used in live VM migration, but it results in longer migration time due to transfer of duplicate memory pages. The proposed algorithm, DedupMR, uses MapReduce to perform parallel deduplication of memory pages, resulting in reduced migration time and downtime[13]. The MapReduce technique processes data in parallel, and duplicated pages were reduced by up to 29% in this study.

Human re-identification (Re-ID) is gaining attention for recognizing suspicious people in security camera footage[14]. However, individual Re-ID poses a security risk for innocent individuals. The proposed technique uses a Haar cascade to ensure confidentiality and safety while preserving individual re-identification in privatized camera videos. This approach addresses computational and memory expenses while ensuring individual safety.

The ResNet50 (a human detection and classification algorithm using Faster R-CNN) is optimized using stochastic gradient descent with momentum (SGDM) for better accuracy and faster training time[15]. This algorithm is important for applications such as self-driving cars, surveillance systems and gender classification[15].

The paper offers a method for human re-identification that can be used to spot suspects in security camera footage. However, this method might jeopardize the privacy of uninvolved parties who appear in the footage. Suggestions for confidentiality-preserving methods that ensure the safety of identified people while protecting their privacy in order to address this[16].

Multimodal biometric systems are more reliable and secure than unimodal systems because they may utilise more than one biometric trait to identify a person[17]. A novel Enhanced Local Line Binary Pattern (ELLBP) approach for extracting ear and fingerprint features is suggested, which enhances identification rate and gives a more reliable multimodal system[17]. Experiment findings using publicly available databases demonstrate that the novel strategy outperforms previous methods, including unimodal systems[17].

A Virtual Private Network (VPN) permits users to make connections to a network securely without exposing their IP address[18]. This technology was initially developed to connect remote users to a secure institutional network. VPN is widely used, but few studies have been conducted on it, and it is often used for illegal activities in the cyber world. Hackers and crackers use VPN to remain anonymous while committing crimes, making it difficult for cyber security experts to track them[18]. The increased usage of multimedia contents has led to the creation of large image and video databases[19]. Content-Based Image Retrieval (CBIR) systems have been developed to efficiently search through these databases, but are limited by the semantic gap. User feedback, or relevance feedback, can be used to reduce the semantic gap and improve the accuracy of retrieval results[19].

The technique enables better load balancing and lowers migration by computing the impact of VM deployment and selecting the least-cost alternative. When compared to standard techniques, experimental results demonstrate that this strategy enhances resource utilization and load balancing under both constant and variable system loads[20]. The "cloud computing" concept transfers resources from a pool to users based on demand in order to suit their needs[21]. By relocating virtual machines from overcrowded to underloaded hosts, live virtual machine migration can address power consumption and load balancing difficulties. This procedure, also known as VM migration, may be carried out while the virtual machines are still running[21].

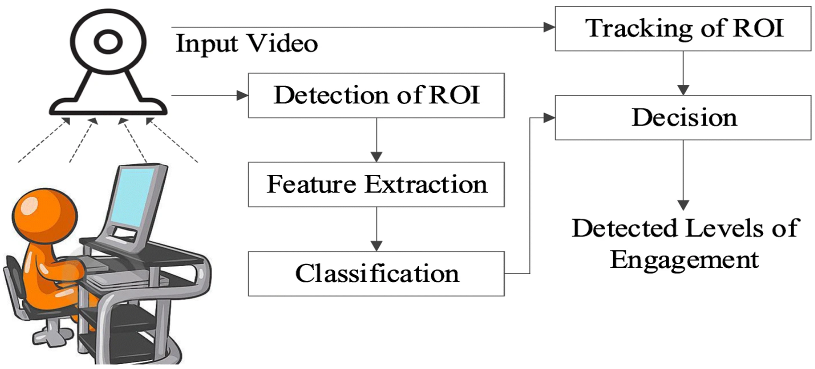

Figure 2.11 A general framework for an engagement detection system based on computer vision.

In order to conduct a thorough review, a general framework as depicted in Figure 2.11, for detecting perceived learner engagement through computer vision methods is proposed. This framework consists of five distinct modules: detection, feature extraction, tracking, classification, and decision. Each module serves a specific function in the process of detecting and analyzing learner engagement.

Dewan, Murshed, and Lin's work "Engagement Detection in Online Learning: A Review" [22] presents a comprehensive assessment of existing research on the detection of student participation in online learning settings. The study discusses numerous techniques to measuring student involvement, such as self-reporting, behavioural observation, and physiological monitoring. The authors also discuss the difficulties in evaluating student involvement in online learning settings, such as the loss of face-to-face interaction, the difficulty in collecting reliable self-reports, and the requirement for real-time monitoring of student engagement. The report also examines the possible advantages of assessing student engagement, such as increasing student outcomes and finding opportunities for instructional improvement. The research examines several ways to detect student interest.

The study "Erson face re-identification using deep learning" [23] outlines a deep learning strategy for matching an individual's face across multiple camera angles. The authors suggest a method for feature extraction that employs a deep convolutional neural network (CNN) and a triplet loss function for model training. On benchmark datasets, the suggested method outperforms various state-of-the-art approaches in terms of accuracy.

2.2. Literature Review Table

The following table, Table 2.1 named Literature survey table is the review for some of the papers that had a similar contribution to the idea but had some shortcomings that have been noted in merits and demerits.

2.1 Literature Survey Table

SI. No.	Research Paper	Merit	Demerit
1	Computer Vision for Attendance and Emotion Analysis in School Settings[4]	It can conduct face detection, face recognition and emotion analysis at beginning and intermediate coding skill levels.	- The software's degree of confidence in distinguishing a person may be affected by distance from the camera.
2	An Integrated Vision System on Emotion Understanding and Identity Confirmation[5]	The facial picture and skeleton are employed for identity verification while the combination of facial landmark action units is used to calculate the emotion understanding.	- Required data for training and subsequent identification is too complicated. - This study uses all the data points including posture and body structures to come to a conclusion.
3	Emotion based mood enhancing music recommendation[6]	- The proposed system is able to process facial images, identify basic emotions and also suggest music that improves the user’s mood.	- The system could include a more variety of emotions to be used as classification classes.

4	Neutral Face Classification Using Personalized Appearance Models for Fast and Robust Emotion Detection[7]	- Reduced model size to work on mobile devices and tablets	- Required data for training and subsequent identification is too complicated.
5	A survey on Facial Emotion Recognition and Classification[8]	- Capable of facial detection, recognition and emotion analysis.	- Requires perfectly aligned cameras for proper data capture.
6	Real - Time Detection if Student Engagement Deep Learning Based System[9]	- During online classes, the system is able to detect the various states of student's attention. - The MobileNet model is used as the foundation for the eye image classification.	- In order to properly capture data, perfectly aligned cameras are required.
7	An intelligent system for monitoring students engagement in large classroom teaching through facial expression recognition[10].	- It is suggested that using facial expression cues rather than behavioral cues to estimate and monitor a student's level of engagement has advantages.	- The model accuracy could use a bit of fine tuning to get better results.
8	Recognising students and detecting Student Engagement with Real-Time Image Processing[11]	- In the course of this research, algorithms were devised for the purpose of face detection and recognition, as all of these efforts yielded fruitful findings, including the evaluation of head posture and the classification of student engagement in the session.	- The proposed algorithm gives amazing results, however the model could still be tweaked to give better results, as always.

9	A new ML - based approach to enhance student engagement in online environment[12]	According to the findings of this study, monitoring student engagement and providing feedback as needed helps students stay on topic, which improves their performance and overall learning.	- The study's two goals to gauge the student involvement levels through the development of effective prediction algorithms and to boost student engagements are both met by the report.
10	Improving state-of-the-art detection of student engagement using Resent and TCN Hybrid Network[13]	For measuring the student involvement in an online environment, a new end-to-end spatiotemporal hybrid architecture, ResNet - TCN, has been developed.	- The effectiveness of the new architecture is clearly visible in the results.
11	Engagement detection in online learning: a review[23]	The article gives a thorough examination of the many methodologies and strategies utilised for engagement detection in online learning, highlighting their advantages and disadvantages.	The article lacks empirical evidence and case studies to support the effectiveness of the various methods discussed. It mostly focuses on theoretical explanations and does not provide practical examples or results of engagement detection in real-world online learning environments.

CHAPTER 3

SYSTEM DESIGN

A vital aspect in the development of any project is system design. It entails the process of defining and refining the system's requirements, identifying the system's components, and producing a thorough plan for how these components will interact to deliver the intended functionality.

The major purpose of system design in a project report is to provide a system blueprint that will guide the development process. This blueprint should be detailed and include all relevant information regarding the system's design, components, and interfaces. The system design should also take into account elements such as performance, scalability, security, and usability, which are crucial to the project's success.

3.1 UML Diagrams

UML diagrams are an important tool for software developers as they allow for visual representation of complex systems and relationships between different components. These diagrams help to ensure that all stakeholders have a common understanding of the system and its functionalities. The different types of UML diagrams include use case diagrams, class diagrams, activity diagrams, sequence diagrams, and state diagrams. Each type of diagram serves a different purpose, such as describing the flow of events in a system or detailing the structure of classes within an application. Overall, UML diagrams are a powerful tool for software development and can greatly aid in the design and implementation of complex software systems.

3.1.1 Activity Diagram:

An activity diagram can be used to show the steps involved in a business or software process. One or more people, pieces of machinery, or pieces of software could carry these out. Activity diagrams are used to document the system's activities, including use cases, business processes, and the actual implementation of those activities like the one shown in Figure 3.1.

The following processes are best described with activity diagrams:

- Case studies and the instructions they provide
- processes in business between users and systems,
- The allowable sequence of component interactions in software
- software programmes

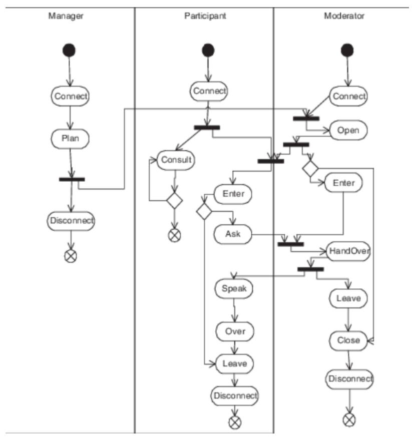

Figure 3.1 Activity diagram of Proposed System

3.1.2 Sequence Diagram:

The sequence diagram, often known as an event diagram, shows how communications move through the system. A variety of dynamic settings can be created more easily as a result. Since it provides the timeline of events, a sequence diagram is useful for outlining the procedures that take place within the application and the model as shown in Figure 3.2 .

The following processes are best described with sequence diagrams:

- To model the intricate interactions between the active components of CNN.
- To model how a network of items interacts in order to realize a use case
- Either broad interactions or particular examples of interactions are simulated.

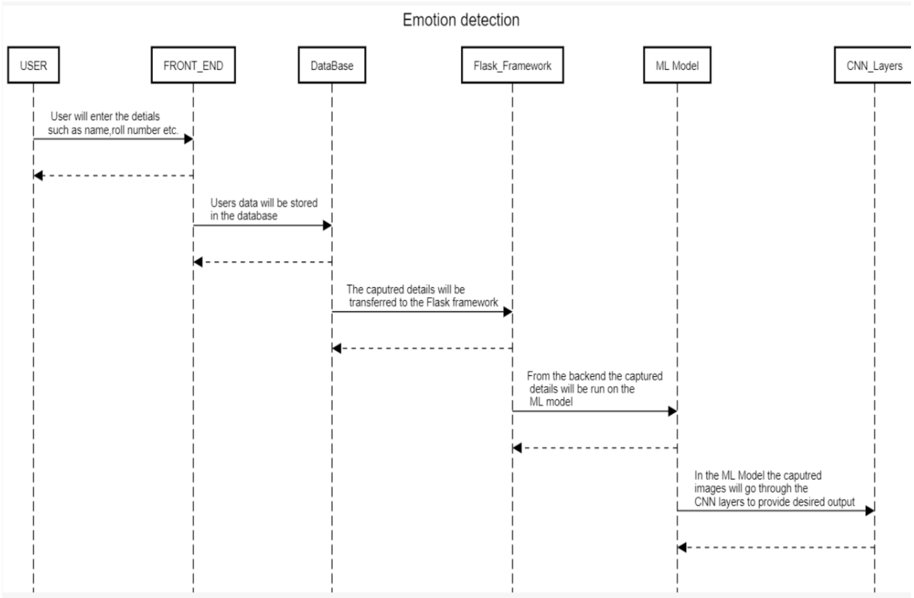

Figure 3.2 Sequence diagram of Proposed System

3.1.3 Communication Diagram:

UML communication diagrams are analogous to sequence diagrams as well as other forms of interaction diagrams in that they show the way objects speak to one another. A communication diagram is an extended form of an object diagram that shows interactions between objects as well as the messages passed between them. The communication diagram also depicts the connections between items and the information they relay.

3.1.4 Interaction Diagram:

Interaction diagrams are used to display the system's interactive behavior. It's challenging to picture the interaction. As a result, the technique calls for the use of a range of models to record the interaction's different facets.

Interaction for expressing the following procedures, diagrams are best:

- to correctly represent the system behavior that is dynamic.
- to explain the system's message flow.
- to explain how things are organized structurally.
- To describe the interactions between objects

3.1.5 Architecture Diagram:

A software system's physical implementation of its component parts is depicted graphically in an architecture diagram. It not only shows the overall organization of the software system but also the connections, constraints, and boundaries that each component has with the others. As shown in Figure 3.3 this is our Architecture diagram.

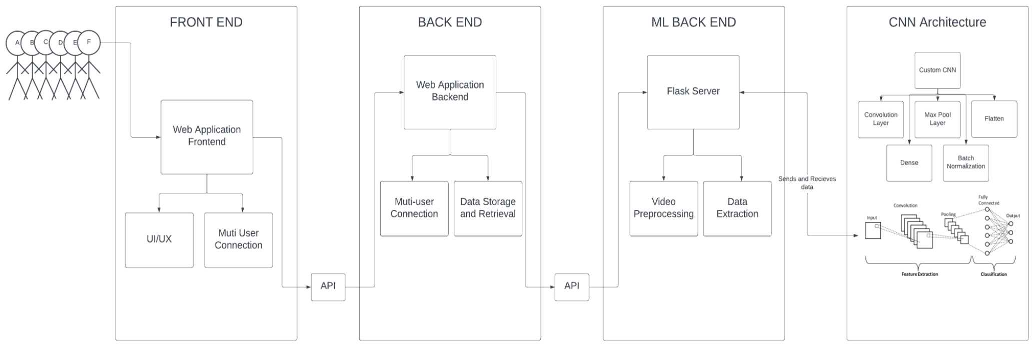

Figure 3.3. Architecture diagram of Proposed System

Figure 3.3 illustrates the architecture flow of the modules in the proposed platform. The platform enables new users to register on the web platform, which is integrated with the backend database through the use of API's. The backend is responsible for storing and retrieving user data in real-time. Additionally, the platform incorporates a Machine Learning (ML) model that is connected to the Flask server, which identifies the objects in the video and extracts relevant data for future reference. The integration of the backend and the ML model ensures efficient and accurate processing of user data and video analysis, providing users with a seamless and effective platform experience. The proposed platform's architecture enables real-time processing, making it an ideal solution for applications that require fast and reliable processing of data.

CHAPTER 4
PROPOSED WORK

The proposed work is a research project that aims to address a specific problem or gap in a particular field. The project will involve a comprehensive study of the problem or gap, and the development of a novel solution or approach to address it. The proposed work may involve the use of various research methods, such as data collection, experimentation, or analysis, to validate the proposed solution or approach.

The outcome of the proposed work will contribute to the existing knowledge in the field and may have practical implications for real-world applications. The proposed work is typically outlined in a research proposal, which provides a detailed description of the research objectives, methodology, and expected outcomes. Here we are discussing the three modules namely, Frontend, Backend and the machine learning model we are looking to integrate.

4.1. Front-End Module

Everything a user interacts with on a website or in a software programme is considered the frontend. The front end is made up of the user interface and design (the appearance) (the feel). Using web languages like HTML, CSS, and JavaScript, the user interface for each of them received the most of the technical work.

The development of the front end experience frequently consists of multiple stages, including the creation of wireframes (basic diagrams of the user flow), prototypes (working versions of the site), and, finally, testing with users. We developed a video conferencing application for this assignment.

People can connect remotely across numerous platforms, devices, and screens using video conferencing. It allows several people to establish an account at the exact same time from any location with an adequate Internet connection. Furthermore, technology provides individuals with fresh methods to interact with those who are crucial to them in both their private and professional lives.

The pages that make up the UI are as follows:

- Users can hold a new meeting on the landing page that we have developed.
- The web conferencing page that offers audio and video streaming for specific sessions as well as related capabilities comes next.
- Join Meeting Using a Unique ID: Users who have created accounts must enter the unique meeting ID in order to join the meeting.
- Participant Camera Feed: It's mostly an app for video conferencing. Other participants are free to share their video feeds, but you are not required to. It will request your consent before using the camera on your computer or other device to broadcast video to the other callers.
- Audio-only conferencing: It is mostly used for video conferencing, but it can also be used for audio-only conferencing. It has many features and improvements that help with video conferencing. Just turn off video when you join a meeting and use the app as you normally would.
- Multi-participants
- video stream switching
- Audio stream on/off switch
- Where users can not only connect with audio and video, but also where users will be able to talk with other users and meet attendees in a room.

During meetings, this is a useful tool for allowing participants to converse with one another without disrupting the speaker who is presenting. Aside from calls, chat may also be used for instant messaging. This is a wonderful approach to convey instructions with other members of the team, as they can refer back to your request at any time.

In addition to the information from a camera, we are going to devise a method for sharing screens during a conference. The function that allows participants to share their screens with other meeting participants on the call is referred to as screen-sharing. You have the option of sharing the screen of your full computer or mobile device, a particular application, a segment of your screen, a whiteboard, and even more. Sharing your screen is a helpful tool for facilitating cooperation inside and across teams. It can help you save time and improve the efficiency of your communication. The ability to share one's screen is one of the features that is utilized the most frequently and is regarded as one of the most convenient functions.

● In addition to this, we are going to develop a recording system that will enable users to take complete audio recordings of meetings. Locally on your computer, you have the ability to record meetings. You can facilitate the sharing of these files with other individuals by uploading them to a file storage service such as Google Drive, Dropbox, or any other comparable alternative.

● Mute individual participants.

● Once we have exited the room using the button labeled "leave," the participant will no longer be present in that particular setting. The link will take you back to the page you originally landed on.

4.2. Back-End Module

Backend development languages handle web application "in the background" processes. It is the code that works the internet-based application, controls client connections, and connects the website to a data server. The front and back ends collaborate to offer the user with the finished product. All the internal working of the online meeting application is handled by the backend. The users of the meeting application have no need to interact with this module, so only the frontend interacts with the backend.

The functionalities of this backend include:

1. Being able to create multiple rooms for online meetings.
2. Allow participants to speak as per their wish.
3. Allow participants to turn on camera as per their wish

The Below Figure 4.1 shows WebRTC, a javascript library developed especially for this, was used to develop this app. Built on top of an existing open standard, WebRTC makes it possible to incorporate real-time communication features into any type of application. Allowing for the sharing of video, speech, and generic data between peers, it aids developers in making effective voice and video communication systems. That's great news since it frees up developers to create innovative new voice and video chat apps. All major platforms and current web browsers have native clients that are compatible with this tech. There is widespread browser support for the WebRTC technologies since they have been adopted as a web standard; these technologies take the form of standard JavaScript APIs (APIs). The same features are made accessible in a library format that may be used by native clients. The WebRTC project is an open-source initiative backed by a wide range of companies, including Apple, Google, Microsoft, and Mozilla.

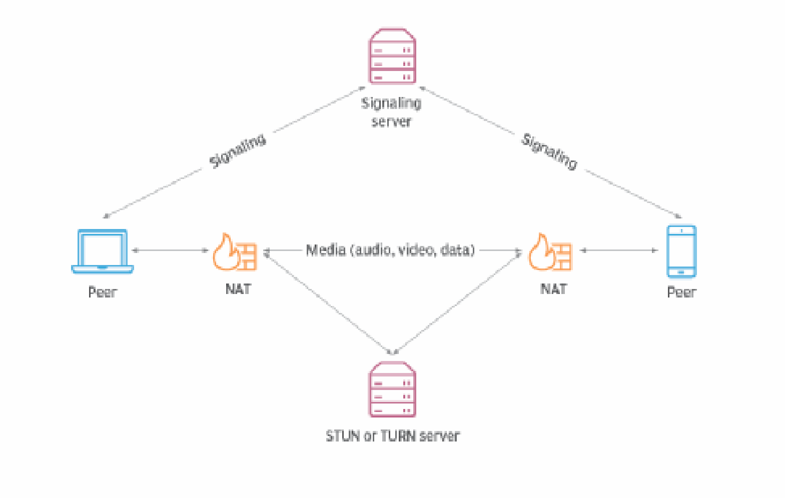

Figure 4.1 WebRTC Working

4.3. Deep Learning Back-end Module

Managing the Deep Learning model is the responsibility of the deep learning backend. It is responsible for processing face data in order to detect engagement. This module never communicates with the front end and only ever talks to the back end. This module is totally separated from the user in every possible way.

The various functionalities that are made available by this module include

Detection of engagement through the application of deep learning models.

1. Delivery of data that is both quick and in real time
2. Capable of pulling face data from different video feeds.

CNNs are frequently used to classify images. The Convolutional Neural Network (CNN) is an intricate system with many moving pieces that serve three main purposes.

4.3.1. Convolution:

Convolution generates the term "convolved feature" which refers to an output feature map that may differ in size and depth from the input feature map, by selecting tiles from the Taking the input feature map and filtering it to find additional features.

Two parameters characterize a convolution:

- Specifications of the selected tiles (typically 3x3 or 5x5 pixels).
- If more filters are used, a more detailed feature map will be produced.

A convolution successfully isolates each matching tile from the filters, which are matrices with the same dimensions as the tile size, over the grid of the input feature map, one pixel at a time in both the horizontal and vertical directions as shown in Figure 4.2.

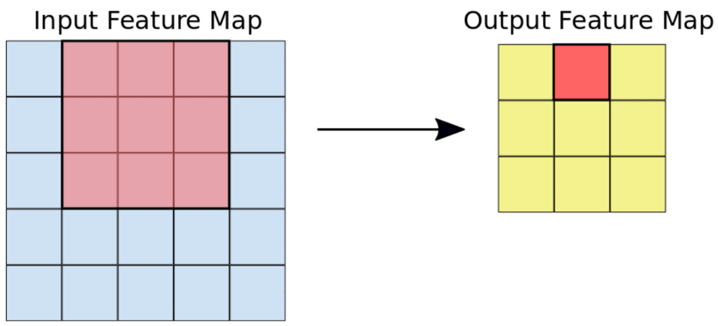

Figure 4.2 Converting Input Feature to Output Feature Map

After performing an element-by-element multiplication between the For each filter-tile pair, the CNN sums the resulting matrix to get a single value. filter matrix and the tile matrix. The resultant values for each filter-tile pair are then output by the convolved feature matrix as shown in Figure 4.3 and 4.4.

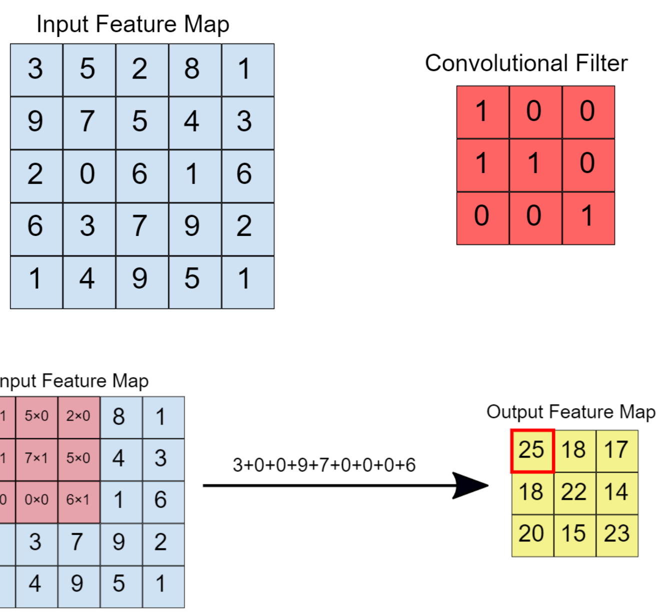

Figure 4.3 and Figure 4.4 Summing up the resultant matrix

In order to extract significant properties (textures, edges, and forms) from the input feature map, the CNN must "learn" the best values to use for the filter matrices during the training process. The more filters applied to the input (also known as the output feature map depth), the more features the CNN can extract. Because filters absorb the majority of CNN's resources, adding more of them increases training time correspondingly. Engineers seek to construct networks that employ the fewest filters possible to extract the information required for correct categorization of images because the added benefit of each successive filter to a network reduces with time.

4.3.2 ReLU:

The CNN performs a Rectified Linear Unit (ReLU) transformation on the convolved feature following each convolution operation to add nonlinearity to the model. ReLU returns x for all x values larger than zero and 0 for all x values less than zero.

4.3.3 Pooling:

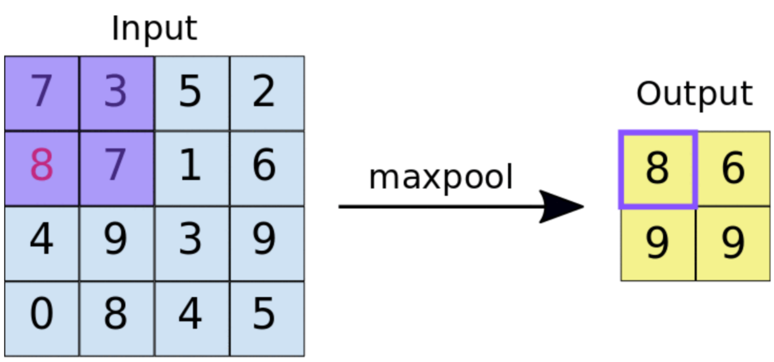

Figure 4.5 Performing Max Pooling

In order to reduce the number of dimensions in the feature map (and hence speed up the processing time), the CNN downsampled the convolved feature after applying ReLU, but without losing any of the essential information about the feature. As shown in Figure 4.5, One often used technique in carrying out this process is the Max pooling algorithm. The function that Max pooling does is similar to that of convolution.By gliding the feature map left and right, you can extract tiles that are a certain size. Each tile's maximum value is sent to a new feature map, and all other values are ignored. There are two inputs needed for maximum pooling operations that is Maximum pooling filter size and stride.

4.4.4 Fully Connected Layer:

A convolutional neural network's final layer or layers may be totally coupled. As shown in Figure 4.6, They must categorize the features that the convolutions acquired. The softmax activation function is often applied in the final completely linked layer of a neural network. This function gives a probability between 0 and 1 for every class that the model is attempting to predict.

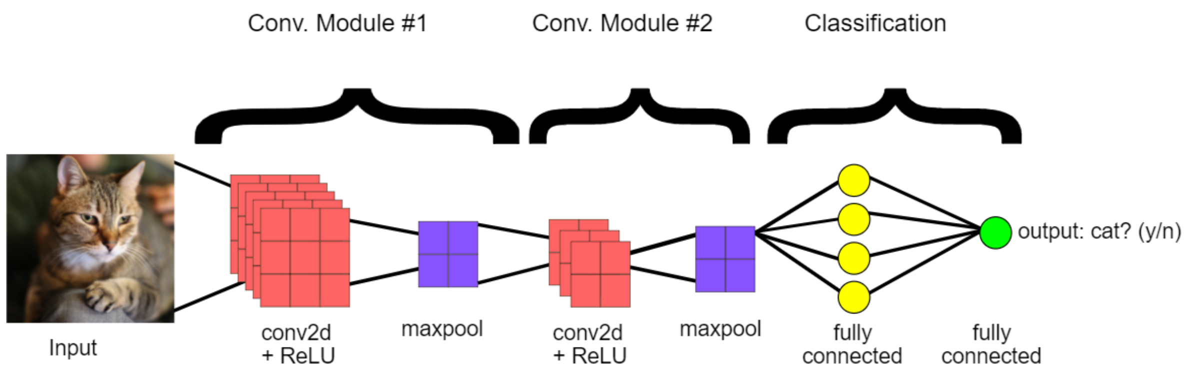

Figure 4.6 Fully Connected Layer

CHAPTER 5

IMPLEMENTATION

We plan to divide the Implementation into three modules. The First would be the Frontend that is seen by the user, The second being the Backend where the servers and the packages are doing the work and third would be the machine learning algorithm.

5.1. Front-End Module Implementation

It is made up of user interface and design, using web languages like React.Js and, the user interface for each of them receives the most of the technical work.

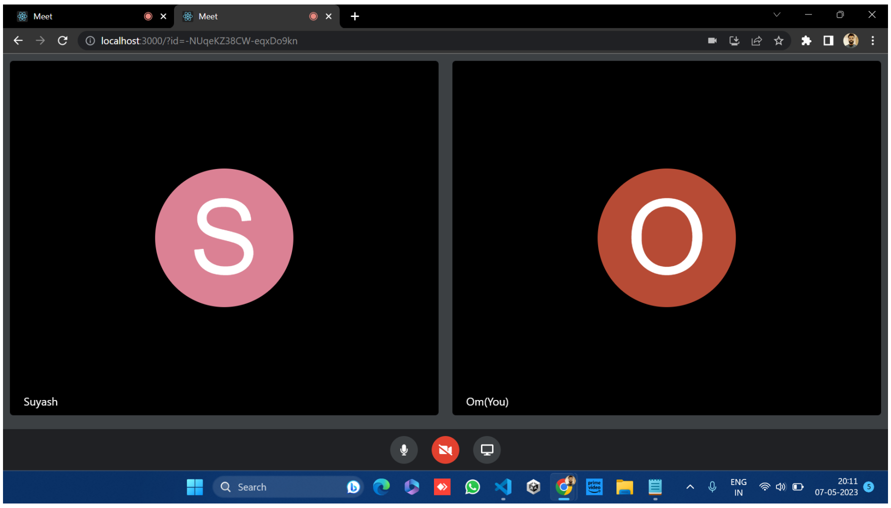

Figure. 5.1 Meet Platform Interface - FrontEnd

The above screenshot named Figure 5.1, the user will be interacting with the platform's interface, which is depicted in the above figure. It is created using frontend tools like React.Js. This process involves several stages including construction of wireframes, prototypes and ultimately user testing.

The development of a web application involves various stages, starting with the design and implementation of the user interface using web technologies such as React.js. The user interface design is a crucial aspect of the process, which requires a significant amount of technical work to ensure usability and effectiveness. The design phase includes the construction of wireframes and prototypes that help to visualize the application's layout and functionality. These mockups are evaluated through user testing to assess their effectiveness and identify areas for improvement. The iterative process of design and testing continues until the final design is developed, which can be implemented and integrated with the application's functionality. Overall, the design of the user interface is a complex and critical aspect of web application development, requiring expertise in web technologies, design principles, and user experience design.

5.2. Back-End Module Implementation

The meet application's internal functionality is entirely handled by the backend, which the meet attendees don't interact with, and only the front-end connects to the backend. The application employs WebRTC, a JavaScript library that provides real-time communication capabilities for custom-made applications. The WebRTC technology utilizes a variety of protocols and APIs, such as STUN, TURN, and ICE, to facilitate real-time communication between browsers. The facial feature detection model illustrated in the above figure will be integrated into our meeting platform, assisting in platform analysis. The convolutional neural network-based model, which can precisely recognise facial features, will be employed to measure student involvement throughout the online learning sessions. Furthermore, the application's front-end is built using web languages such as React.js, and it requires significant technical work, including wireframe construction, prototyping, and user testing, to develop an optimal user interface.

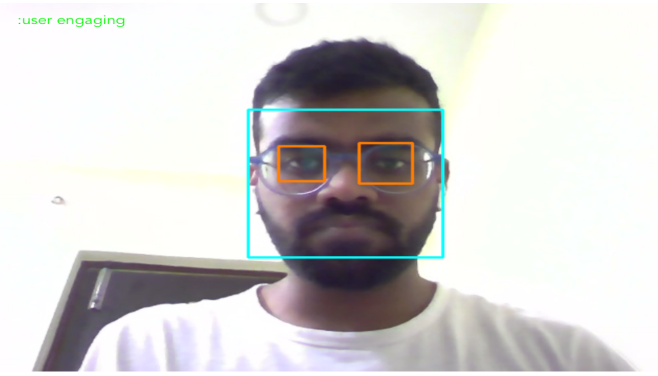

Figure. 5.2 Feature Detection

The running model for a human face's feature detection is shown in the above Figure 5.2. Our meeting platform will integrate this model, which will aid in platform analysis.

5.3. Deep Learning Module Implementation

The engagement detection module is a crucial component of the online learning platform, responsible for accurately detecting and measuring student engagement without the need for user input. This module operates independently of the user and uses advanced machine learning techniques such as Convolutional Neural Networks (CNNs) to classify facial images and extract relevant features for engagement detection. The CNN is a complex neural network architecture that uses multiple layers of convolution, ReLU activation, and pooling to analyze and interpret images. By leveraging this advanced technology, the engagement detection module can provide more accurate and reliable results compared to traditional methods, improving the overall quality of the online learning experience.

```
model = Sequential()

model.add(Conv2D(32, kernel_size=(3, 3), activation='relu', input_shape=(48,48,1)))
model.add(Conv2D(64, kernel_size=(3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Conv2D(128, kernel_size=(3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Conv2D(128, kernel_size=(3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Flatten())
model.add(Dense(1024, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(7, activation='softmax'))
```

Figure 5.3. Part of CNN using Keras framework

The above image in figure 5.3 shows the code that defines the Keras framework to construct a convolutional neural network (CNN). There are multiple layers in the model: 32 filters make up the first layer, which is a 2D convolutional layer, each with a kernel size of 3x3 and activated by the rectified linear unit (ReLU). The input shape of the layer is 48x48x1, meaning the input is a grayscale image with dimensions 48x48. Another 2D convolutional layer with 64 filters, a kernel size of 3x3, and the ReLU activation function makes up the second layer. Following that is a max pooling layer, which takes the maximum value within a 2x2 frame to minimize the dimensionality of the feature maps created by the convolutional layers. A dropout layer is added to help prevent overfitting. This randomly drops out 25% of the neurons in the layer during each training epoch. Two more 2D convolutional layers follow, each with 128 filters and a kernel size of 3x3. The ReLU activation function is used for both.

Two more max pooling layers follow, each with a 2x2 window. Another dropout layer is added to further help prevent overfitting. The previous layer's 2D feature maps are flattened to create a 1D vector that may be utilized as input for a fully linked layer by adding a flatten layer. The flattened layer is followed by a layer with 1024 neurons and the ReLU activation function that is fully connected. To the completely connected layer, another dropout layer is added.

The output of the network is then created by adding a dense layer with 7 neurons and the softmax activation function. The 7 output neurons correspond to the 7 emotions that the network can recognize (neutral, happy, sad, angry, fearful, disgusted, surprised) according to which it decides if the user is engaging.

Overall, this CNN is designed to recognize emotions in facial expressions from grayscale images with dimensions of 48x48.

5.3.1. Implementation of Convolution

In image processing and computer vision, convolution is a potent mathematical operation that is frequently employed. Many contemporary deep learning architectures, especially convolutional neural networks (CNNs), use it as a key building block, where it plays a crucial role in extracting relevant features from images.

At its core, convolution involves the use of small filters, also known as kernels or masks, to extract relevant information from an input image. These filters are typically small matrices of numbers that are applied to the image pixel by pixel. The filter is centered on a different pixel at each step, and the values of the filter are multiplied by the corresponding pixel values in the image element-by-element. The resulting products are summed, and the sum is assigned to the output pixel at the same location as the center of the filter. Every pixel in the image goes through this procedure once more to create a new image known as a feature map.

The purpose of convolution is to extract relevant features from the input image that are useful for a particular task, such as object detection, image classification, or facial recognition. The filters used in convolution are carefully designed to capture certain visual patterns or structures in the image, such as edges, corners, textures, or shapes. By applying different filters to the same input image, we can extract multiple sets of features that represent different aspects of the image.

One of the key advantages of convolution is its ability to capture spatial information in the input image. Because the filter is applied to each pixel in the image, the resulting feature map preserves the spatial structure of the original image. This is especially helpful for jobs like object detection when pinpointing the exact location of an object in a picture is crucial. By applying convolution to multiple layers of the image, we can capture increasingly complex patterns and structures that are relevant to the task.

Convolution is also a computationally efficient operation, which makes it well-suited for deep learning applications. Because the same filter is applied to multiple pixels in the image, we can perform convolution using a sliding window approach that shares the same filter parameters across all the pixels. This reduces the number of parameters that need to be learned, making convolution an efficient way to extract features from large images.

In deep learning, convolution is typically used in conjunction with other operations, such as pooling and activation functions, to create convolutional neural networks (CNNs). Many layers of convolutional, pooling, and activation functions are used in a typical CNN before one or more fully linked layers are added. The purpose of the convolutional layers is to extract relevant features from the input image, while the pooling layers reduce the size of the feature maps by downsampling them. By introducing non-linearity to the network, the activation functions enable the network to learn intricate representations of the input data.

One of the key challenges in using convolution in deep learning is the choice of filter parameters, such as the size of the filter and the number of filters to use. These parameters can have a significant impact on the performance of the network, and choosing the optimal values requires careful tuning and experimentation. Additionally, convolution can be sensitive to variations in lighting, rotation, and other factors that can affect the appearance of the input image. To address these challenges, researchers have developed a wide range of techniques, such as data augmentation, regularization, and transfer learning, to improve the robustness and accuracy of convolutional neural networks.

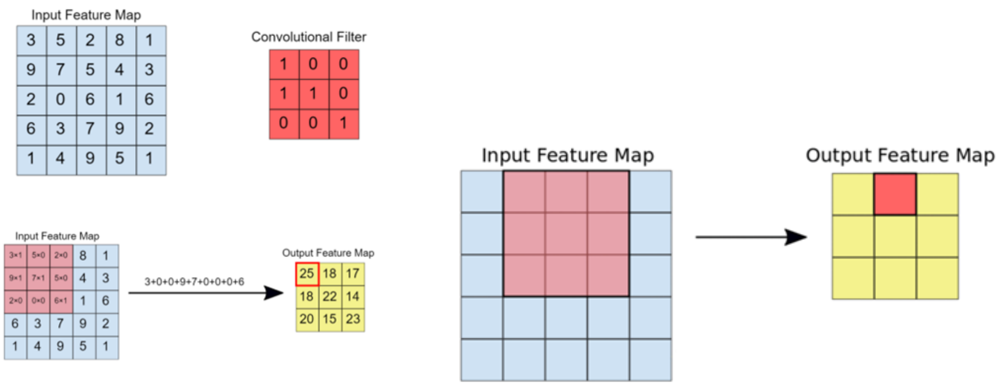

Figure 5.4. Converting Input Feature to Output Feature Map and summing up the resultant matrix.

This is the resultant after performing an element-by-element multiplication between the filter matrix and the tile matrix for each filter-tile pair, the CNN then sums the resulting matrix to yield a single value as shown in Figure 5.4. The resultant values for each filter-tile pair are then output by the convolved feature matrix.

5.3.2. Implementation of ReLU

In order to provide nonlinearity to the model, the CNN performs a Rectified Linear Unit (ReLU) transformation on the convolved feature after each convolution operation. ReLU yields x for all values of x greater than zero and 0 for all values of x lower than zero.

5.3.3. Implementation of Pooling

In order to reduce the number of dimensions in the feature map (and hence speed up the processing time), the CNN downsampled the convolved feature after applying ReLU, but without losing any of the essential information about the feature. One often used technique in carrying out this process is the Max pooling algorithm.

As shown in Figure 5.5, The function that Max pooling does is similar to that of convolution. Extract tiles of a specified size by gliding the feature map left and right. The maximum value for each tile is sent to a new feature map while all other values are discarded.

There are two inputs needed for maximum pooling operations that is Maximum pooling filter size and stride.

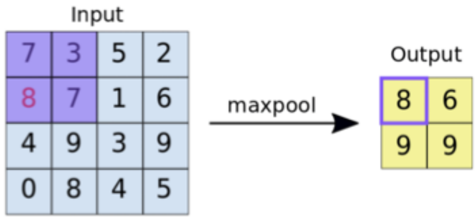

Figure 5.5. Depicts the procedure of max pooling

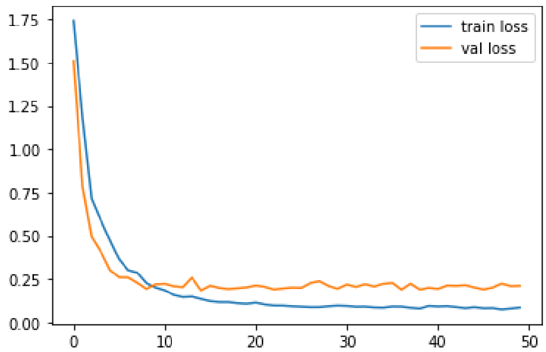

Figure 5.6. Train Vs Value LossGraph

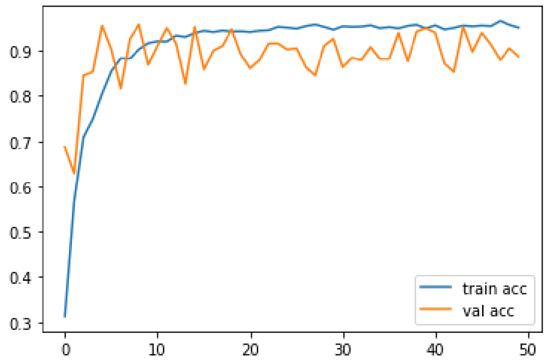

Figure 5.7. Train Vs Value Accuracy Graph

The first plot, namely Figure 5.6 depicts the model's training and validation loss. The plt.plot() function is used to plot the training loss values from the model_info.history dictionary, which contains the training loss values. The label argument is used to define the plot's legend. The same function, but with a different label, is used to visualize the validation loss values from the same dictionary. Plt.legend() displays the legend in the plot, and plt.show() displays the plot in a window. Finally, plt.savefig() creates an image file from the plot.

The second plot, namely Figure 5.7, depicts the model's training and validation accuracy. The plt.plot() function, like the first plot, is used to plot the training and validation accuracy values from the model_info.history dictionary. The label argument is used to define the plot's legend. Plt.legend() displays the legend in the plot, and plt.show() displays the plot in a window. Finally, plt.savefig() creates an image file from the plot.

Overall, these plots can help you understand how the model is performing during training and validation, and can give insight into whether the model is overfitting or underfitting.

CHAPTER 6

TESTING

Testing is an important part of software development since it involves reviewing and confirming the quality and functionality of software applications. The purpose of testing is to uncover faults, defects, or other issues that may influence the application's performance, functionality, and user experience. We will explore the necessity of testing, the many types of testing, and the best practices for performing good testing in this white paper.

6.1 The Importance of Testing:

Testing is important in software development because it guarantees the application fulfills its stated specifications and operates as expected. The following are some of the benefits of testing:

Improved Quality: Testing helps identify and fix defects, errors, and bugs in the application, leading to an improved overall quality.

Reduced Costs: Testing helps identify and fix issues early in the development cycle, reducing the costs associated with fixing defects in the later stages of development.

Improved User Experience: Testing helps identify and fix issues that could affect the user experience, leading to a more positive user experience.

Increased Trust: Testing helps increase the trust and confidence in the application, both for the end-users and the development team.

6.2 Types of Testing

There are several types of testing that can be conducted during the software development lifecycle. Each type of testing has a specific goal and objective and is essential in ensuring the overall quality of the application. The following are some of the different types of testing:

Unit testing is testing individual components or units of code in isolation to ensure that they work as expected. Developers frequently perform automated unit testing.

Integration testing entails determining how different components or modules of an application interact with one another in order to ensure proper integration. Integration testing is typically conducted by developers or quality assurance (QA) professionals. System Testing: System testing is testing the complete system to confirm that it works properly and meets the specified standards. QA personnel are often in charge of system testing.

Acceptance testing involves evaluating the application employing data from the real world to ensure that it fulfills the expectations and requirements of the end users. End-users or QA experts often perform acceptance testing.

Performance testing involves assessing an application's capacity and efficiency under various scenarios, such as heavy loads or high traffic. QA experts are often in charge of testing for performance.

White box testing involves testing the internal workings of the software, with knowledge of its underlying code, to ensure that it is functioning correctly. Testers typically use tools such as code reviews and unit tests to perform white box testing.

In contrast, black box testing involves testing the software's external functionality, without any knowledge of its internal workings. Testers typically perform black box testing by using the software as an end user would, testing the software's functionality against predefined requirements and scenarios.

Alpha testing and beta testing are two types of software testing performed during the software's release cycle. Alpha testing is typically performed by a select group of testers, often the developers themselves, to identify and fix any issues before the software's release to a wider audience.

Beta testing is performed by a larger group of users who test the software in a real-world environment, providing feedback on its functionality and identifying any issues that may have been missed during the alpha testing phase. Beta testing provides valuable feedback to developers, allowing them to make any necessary changes before the software's final release.

6.3 Best Practices for Conducting Effective Testing

Effective testing requires careful planning, execution, and reporting. The following are some best practices for conducting effective testing:

Define Testing Objectives: Clearly define the testing objectives and goals, including the types of testing to be conducted, the expected outcomes, and the criteria for success. Create a Test Plan: Create a test plan that outlines the testing process, including the testing scope, the test cases, and the testing environment.

Use Automation: Use automation tools to automate repetitive testing tasks, such as unit testing, to increase efficiency and accuracy. Use Real-World Data: Use real-world data to simulate real-world scenarios and ensure that the application meets the end-users' expectations and requirements.

Test Early and Often: Test early and often throughout the software development lifecycle to identify and fix issues before they become more complex and costly to fix. Use Multiple Testing Approaches: Use multiple testing approaches, including manual and automated testing, to ensure that all aspects of the application are tested thoroughly.

Involve Stakeholders: Involve stakeholders, including developers, QA professionals, and end-users, in the testing process to ensure that the application meets everyone's needs and expectations.

6.4 Test Objective and Results

For this project, the practice of unit testing has been employed. Unit testing is a method of software testing that involves the testing of individual components or units of a software system to ensure that each one performs as expected. By isolating and testing these components independently, developers can identify and fix any issues early in the development process, before they become more complex and difficult to solve. Unit testing is an important part of the software development life cycle and helps to ensure that software products are reliable, stable, and perform as intended.

We have employed the method of unit testing as the three modules that we have created must be tested individually before being integrated as one complete application to know the flow of data.

The main objective of this test is to check whether we are able to create multiple instances using a single link provided in the meet platform. To do this we will use the same link that will be given while hosting the server. We also have an additional objective to see if the camera is able to take the user video in real time.

The success of our test objective to create multiple instances in the meet application with the simulation of multiple devices using multiple web tabs. The figures depict multiple instances of the meet application, each with a unique username and profile picture, indicating the successful creation of multiple instances. The test results demonstrate the ability of the application to handle multiple instances and simulate multiple devices, which is crucial for ensuring the scalability and reliability of the platform. Overall, the test results are promising and indicate that the meet application is capable of meeting the demands of a large user base.

Our next objective aligns with the Deep learning image detection model we have created. The main objective in this unit test is to check whether we are able to detect if an expression is being detected or not. If it is detecting facial expression, we also have to check if it's detecting the correct expression. We also have an additional objective to check whether the user is engaging or not using the detection of facial expressions.

We can conclude that the objectives of our test have been met. The results indicate that the system was able to function efficiently and effectively in multiple instances, as well as with multiple devices using multiple web tabs. Also the image detection algorithm was able to detect multiple emotions in real-time. By reviewing these figures, we can conclude that the implementation of the meet application has been successful in achieving its intended goals.

CHAPTER 7

RESULTS AND DISCUSSION

Online meeting platforms have grown in popularity recently, particularly with the introduction of remote employment and social seclusion measures as a result of the COVID-19 epidemic. Following are some hypotheses on online meeting platforms. According to the Technology Acceptance Model (TAM), perceived utility and perceived user-friendliness have an impact on users' acceptance of technology. Users are more likely to accept and keep utilizing online meeting platforms if they feel them to be helpful and simple to use.

7.1 Screenshots - Meet Platform

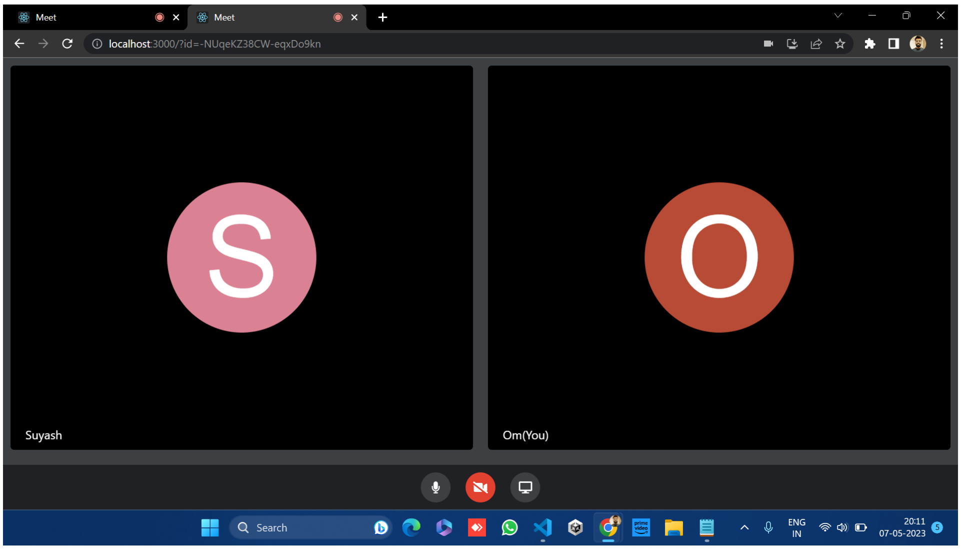

Figure 7.1. Screenshot of meet application with two users

The Figure 7.1 and Figure 7.2, shows the meeting platform with multiple instances created within the meet. Here, we used React.Js, Javascript, HTML and CSS. The code imports the required libraries, including NPM, Node.Js, WebRTC for the working of the back-end module.

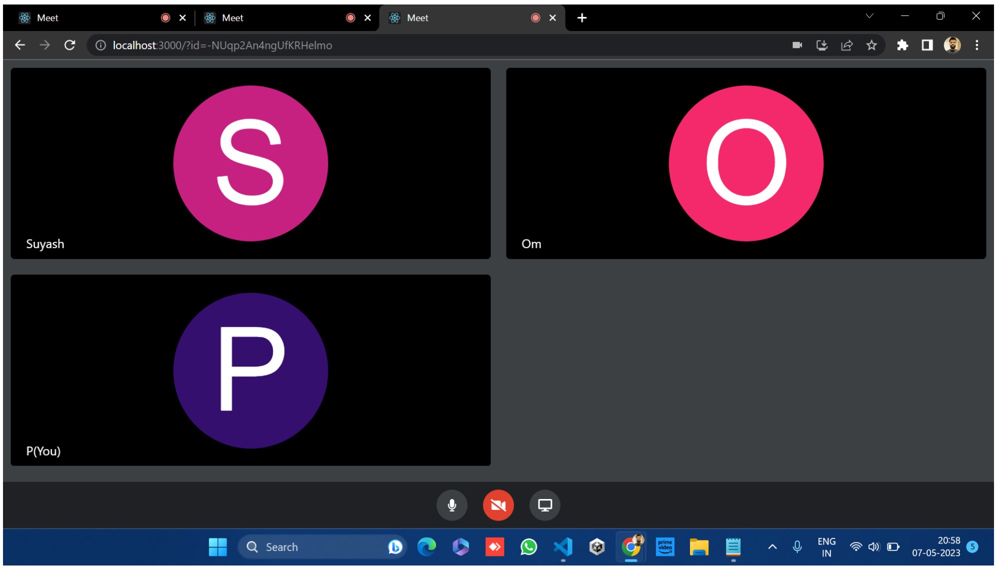

Figure 7.2 Screenshot of meet application with three users

7.2 Model Screenshots

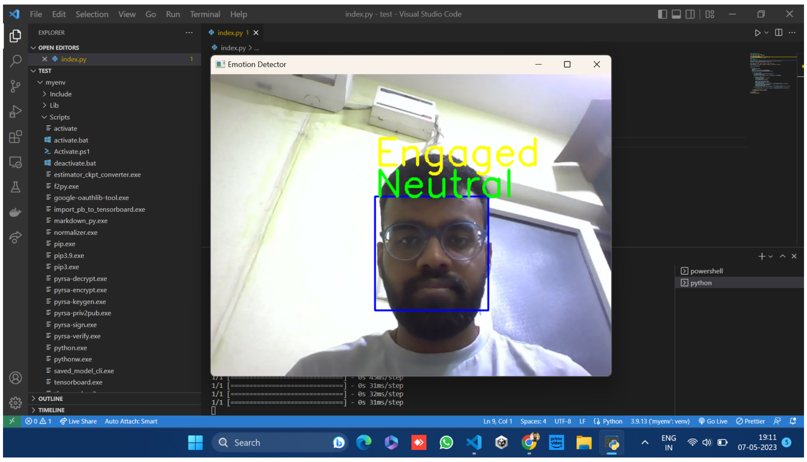

Figure 7.3 Facial Detection as Neutral / Engaged

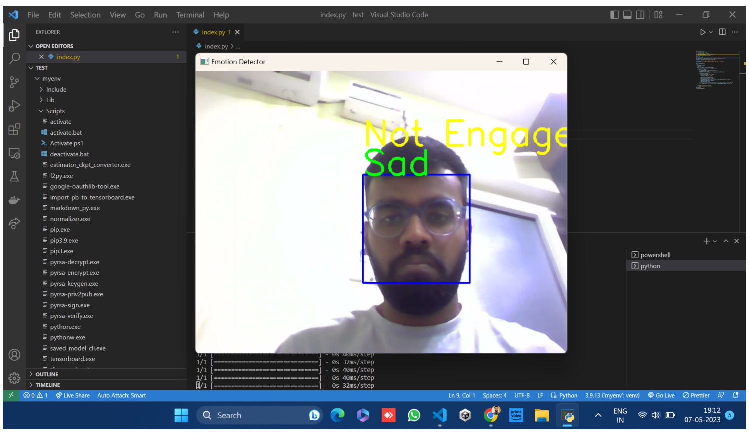

Figure 7.4 Facial Detection as Sad / Not Engaged

The model is trained to classify facial expressions into one of five emotion categories: Angry, Happy, Neutral, Sad, and Surprise as shown in Figures 7.3, 7.4, 7.5. The code then initializes a video capture object using OpenCV to capture video frames from a webcam or other video source.

The pre-trained CNN model is then used to predict the emotion label for the cropped face image. The predicted emotion label is then overlaid onto the video frame, along with an additional label indicating whether the person is "Engaged" or "Not Engaged" based on the predicted emotion label.

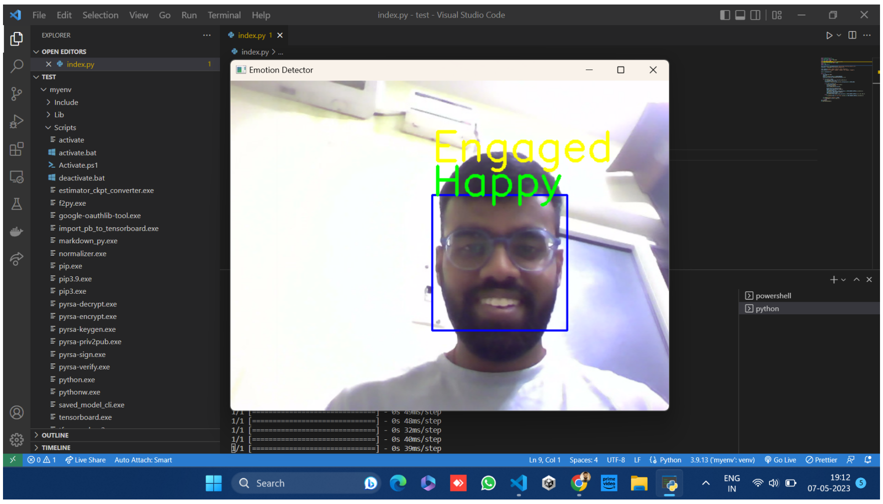

Figure 7.5 Facial Detection as Happy / Engaged

While the code provides a basic implementation of real-time facial emotion detection, it has some limitations. For instance, the code only detects four emotions (happiness, sadness, wrath, and neutrality) out of the many possible emotions that humans can express. Additionally, the accuracy of the emotion detection may vary depending on lighting conditions, camera quality, and the facial expressions of the person being detected.

Overall, the code presented here provides a starting point for developers and researchers interested in real-time facial emotion detection. With further improvements and modifications, this code can be used to develop more robust and accurate facial emotion detection systems for a variety of applications.

CHAPTER 8

CONCLUSION

Throughout the course of this project, we had reviewed the prior research papers. As a result of the work that we did, we now have the knowledge necessary to develop our very own deep learning model for engagement detection. We have built different architectures of deep learning models to detect the engagement of the student in the class. In addition to that, we have built a web application that simulates the operation of a platform for holding online meetings.

The goal of this project is to accomplish the integration of the model into the application so that real-time data can be obtained from online meetings. The identification of emotions served as the foundation for our first set of models, and we have used them to develop a metric for gauging levels of user engagement. This strategy was quickly elevated to a higher level when an original engagement detection convolutional neural network was conceived of as a potential improvement. The emotion detection model will play an important part for the fact that we are now able to make a decision that is better informed because of having more data. So we have finally come up with different architectures of deep learning models which are used to detect the engagement level of the student and have also built a virtual meet application for holding the online meetings. Thus, in this project we can measure the attentiveness of the student in a virtual meeting. The conclusive result of this comparative study comes out to be highly insightful.

All the papers that were considered for this study came with their own unique set of advantages and disadvantages. Pro and cons being the two sides of the same coin, this was the expected outcome. With the end of this comparison, we have successfully been able to determine our next actions to be able to build a fully functional Student Engagement Detection system.

The study's conclusive findings have shed light on the various benefits and drawbacks of the articles considered. It is not surprising that each article has its unique set of advantages and limitations, as these two aspects are two sides of the same coin. However, this comparison has been valuable in determining the next steps required to develop a system that can accurately detect student engagement. By analyzing the various articles, we can now identify the strengths and weaknesses of different approaches to detecting student engagement. This information is invaluable in informing the development of a new system that can overcome the limitations of previous methods and capitalize on their benefits. With this knowledge, we can better design a system that accurately detects student engagement, improving the overall learning experience for students.

Furthermore, the process of comparing these articles has allowed us to gain a deeper understanding of the complexity involved in detecting student engagement. It is not a straightforward task, and there are numerous factors to consider, including the type of engagement being measured, the technology used, and the environment in which it is deployed. By acknowledging and understanding these factors, we can develop a more nuanced approach to detecting student engagement.

CHAPTER 9

FUTURE WORK

It is possible to enhance this model in the future to recognize a wider range of emotions beyond the current seven that it is capable of detecting. By expanding the model's capabilities to identify a more extensive range of emotions, it will improve its accuracy in real-world settings. This advancement is particularly useful because it can be challenging to accurately recognize emotions in complex scenarios, and the ability to detect a broader range of emotions will help to address this issue. Additionally, there is the potential to integrate this model into a larger piece of software that can evaluate its performance in practical settings. By incorporating the model into a comprehensive system, it will be possible to test and refine its effectiveness in various scenarios, allowing for continuous improvements and refinements. This approach will ensure that the model is continually adapting and improving, ultimately leading to better accuracy and effectiveness in recognizing emotions.

As technology evolves, there are opportunities to add more features and capabilities to the model. Advancements in machine learning and artificial intelligence could detect subtle changes in facial expressions and body language, improving accuracy. New sensors and data sources could expand the model's applications. Integrating deep learning into a WebRTC backend for a react.js meet application could improve real-time emotion detection in video conferences.

By integrating deep learning algorithms, the application could recognize complex emotional cues and patterns in real-time, leading to more accurate and nuanced emotional recognition. This integration would be particularly beneficial in virtual meetings where participants' emotions can be challenging to interpret, given the limited visual and auditory cues available.

REFERENCES

[1] S. Deniz et al., "Computer Vision for Attendance and Emotion Analysis in School Settings," 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), 2019, pp. 0134-0139, doi: 10.1109/CCWC.2019.8666488.

[2] Y. -Y. Ou, A. -C. Tsai, J. -F. Wang and P. -C. Lin, "An integrated vision system on emotion understanding and identity confirmation," 2017 International Conference on Orange Technologies (ICOT), 2017, pp. 205-209, doi: 10.1109/ICOT.2017.8336123.

[3] M. Moolchandani, S. Dwivedi, S. Nigam and K. Gupta, "A survey on: Facial Emotion Recognition and Classification," 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), 2021, pp. 1677-1686, doi: 10.1109/ICCMC51019.2021.9418349.

[4] A. V. Iyer, V. Pasad, S. R. Sankhe and K. Prajapati, "Emotion based mood enhancing music recommendation," 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), 2017, pp. 1573-1577, doi: 10.1109/RTEICT.2017.8256863.

[5] P. Chiranjeevi, V. Gopalakrishnan and P. Moogi, "Neutral Face Classification Using Personalized Appearance Models for Fast and Robust Emotion Detection," in IEEE Transactions on Image Processing, vol. 24, no. 9, pp. 2701-2711, Sept. 2015, doi: 10.1109/TIP.2015.2421437.

[6] Ahmed, Zeyad Abdulhameed & Jadhav, Mukti & Al-madani, Ali & Tawfik, Mohammed & Alsubari, Saleh & Alshareef, Ahmed. (2022). Real-Time Detection of Student Engagement: Deep Learning-Based System. 10.1007/978-981-16-2594-7_26.

[7] Pabba, Chakradhar & Kumar, Praveen. (2022). An intelligent system for monitoring students' engagement in large classroom teaching through facial expression recognition. Expert Systems. 39. 10.1111/exsy.12839.

[8] Apicella, Andrea & Arpaia, Pasquale & Frosolone, Mirco & Improta, Giovanni & Moccaldi, Nicola & Pollastro, Andrea. (2021). EEG-based Measurement System for Student Engagement Detection in Learning 4.0. 10.21203/rs.3.rs-944837/v1.

[9] Uçar, Mustafa & Özdemir, Ersin. (2022). Recognizing Students and Detecting Student Engagement with Real-Time Image Processing. Electronics. 11. 10.3390/electronics11091500.

[10] Ayouni S, Hajjej F, Maddeh M, Al-Otaibi S. A new ML-based approach to enhance student engagement in online environment. PLoS One. 2021 Nov 10;16(11):e0258788. doi: 10.1371/journal.pone.0258788. PMID: 34758022; PMCID: PMC8580220.

[11] Ali Abedi, Shehroz S. Khan: “Improving state-of-the-art in Detecting Student Engagement with Resnet and TCN Hybrid Network”, 2021; [http://arxiv.org/abs/2104.10122 arXiv:2104.10122]..

[12] Ahmed, Zeyad Abdulhameed & Jadhav, Mukti & Al-madani, Ali & Tawfik, Mohammed & Alsubari, Saleh & Alshareef, Ahmed. (2022). Real-Time Detection of Student Engagement: Deep Learning-Based System. 10.1007/978-981-16-2594-7_26.

[13] Pabba, Chakradhar & Kumar, Praveen. (2022). An intelligent system for monitoring students' engagement in large classroom teaching through facial expression recognition. Expert Systems. 39. 10.1111/exsy.12839.

[14] Y. -Y. Ou, A. -C. Tsai, J. -F. Wang and P. -C. Lin, "An integrated vision system on emotion understanding and identity confirmation," 2017 International Conference on Orange Technologies (ICOT), 2017, pp. 205-209, doi: 10.1109/ICOT.2017.8336123.

[15] S. Deniz et al., "Computer Vision for Attendance and Emotion Analysis in School Settings," 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), 2019, pp. 0134-0139, doi: 10.1109/CCWC.2019.8666488.

[16] P. Chiranjeevi, V. Gopalakrishnan and P. Moogi, "Neutral Face Classification Using Personalized Appearance Models for Fast and Robust Emotion Detection," in IEEE Transactions on Image Processing, vol. 24, no. 9, pp. 2701-2711, Sept. 2015, doi: 10.1109/TIP.2015.2421437.

[17] A. V. Iyer, V. Pasad, S. R. Sankhe and K. Prajapati, "Emotion based mood enhancing music recommendation," 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), 2017, pp. 1573-1577, doi: 10.1109/RTEICT.2017.8256863.

[18] Pilkington, Marc. (2016). Blockchain Technology: Principles and Applications.

[19] Apicella, Andrea & Arpaia, Pasquale & Frosolone, Mirco & Improta, Giovanni & Moccaldi, Nicola & Pollastro, Andrea. (2021). EEG-based Measurement System for Student Engagement Detection in Learning 4.0. 10.21203/rs.3.rs-944837/v1.

[20] Uçar, Mustafa & Özdemir, Ersin. (2022). Recognizing Students and Detecting Student Engagement with Real-Time Image Processing. Electronics. 11. 10.3390/electronics11091500.

[21] Ayouni S, Hajjej F, Maddeh M, Al-Otaibi S. A new ML-based approach to enhance student engagement in an online environment. PLoS One. 2021 Nov 10;16(11):e0258788. doi: 10.1371/journal.pone.0258788. PMID: 34758022; PMCID: PMC8580220.

[22] Dewan, M.A.A., Murshed, M. & Lin, F. Engagement detection in online learning: a review. *Smart Learn. Environ.* 6, 1 (2019). https://doi.org/10.1186/s40561-018-0080-z

[23] Turlapati, Malleswari & G, Dr. Vadivu & Malathi, D.. (2014). Live Virtual Machine Migration Techniques-A Technical Survey. Springer India, Intelligent Computing Communication and Devices. 308. 10.1007/978-81-322-2012-1_32.

[24] Turlapati, Malleswari & G, Dr. Vadivu. (2017). Deduplication of VM memory pages using MapReduce in live migration. ARPN Journal of Engineering and Applied Sciences. 12. 1890-1894.

[25] T.BHASKAR REDDY, B.MURALI DHAR NAIDU, S.USHASUKHANYA. (2020). PERSON FACE RE-IDENTIFICATION USING DEEP LEARNING APPROACH. International Journal of Advanced Science and Technology, 29(6s), 2290-2295.

[26] Ms. Avani Dungarwal, Mrs. Ushasukhanya, Mr. Parth Jain. (2020). OPTIMIZATION OF DEEP LEARNING FOR HUMAN DETECTION. International Journal of Advanced Science and Technology, 29(6s), 1873 - 1883. Retrieved from http://sersc.org/journals/index.php/IJAST/article/view/9350.

[27] T.BHASKAR REDDY, B.MURALI DHAR NAIDU, S.USHASUKHANYA. (2020). PERSON FACE RE-IDENTIFICATION USING DEEP LEARNING APPROACH.

[28] Malathy C,Sadiq M.A.K, Annapurani K,"A new feature extraction technique for person identification using multimodal biometrics", Research Journal of applied sciences,Engineering and technology,8(12),pp 1492-1497,2014.

APPENDIX

CODING

In the discipline of computer science, the word "code" is used to refer to the individual algorithms that come together to form a computer programme. These algorithms are made up of symbols that have been extracted from a source alphabet, and they reflect a set of rules that specify the activities that the software is intended to carry out.

Meet Platform Firebase Connection

```
import firebase from "firebase";

var firebaseConfig = {

  apiKey: "AIzaSyALDex0b7Xk6_qzu_HpKQwRxMDitXLrq4c", // Add
API Key

  databaseURL:

"https://meetfinallogin-default-rtdb.asia-southeast1.firebased
atabase.app", // Add databaseURL

};// Initialize Firebase

firebase.initializeApp(firebaseConfig);

export const db = firebase;

var firepadRef = firebase.database().ref();



export const userName = prompt("What's your name?");

const urlparams = new URLSearchParams(window.location.search);

const roomId = urlparams.get("id");
```

```
if (roomId) {
  firepadRef = firepadRef.child(roomId);
} else {
  firepadRef = firepadRef.push();
  window.history.replaceState(null, "Meet", "?id=" +
firepadRef.key);
}
export default firepadRef;
```

FACIAL EXPRESSION RECOGNITION MODEL

```
import tensorflow as tf
from tensorflow import keras
from keras.models import load_model
from time import sleep
from tensorflow.keras.utils import img_to_array
from keras.preprocessing import image
import cv2
import numpy as np
face_classifier=cv2.CascadeClassifier('haarcascade_frontalface
_default.xml')
classifier = load_model('EmotionDetectionModel.h5')
```

```
class_labels1=['Angry','Happy','Neutral','Sad','Surprise']

class_labels2=['Not Engaged','Engaged','Engaged','Not
Engaged','Engaged']

cap=cv2.VideoCapture(0)

while True:

    ret,frame=cap.read()

    labels=[]

    gray=cv2.cvtColor(frame,cv2.COLOR_BGR2GRAY)

    faes=face_classifier.detectMultiScale(gray,1.3,5)

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

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

        roi_gray=gray[y:y+h,x:x+w]

roi_gray=cv2.resize(roi_gray,(48,48),interpolation=cv2.INTER_A
REA)



        if np.sum([roi_gray])!=0:

            roi=roi_gray.astype('float')/255.0

            roi=img_to_array(roi)

            roi=np.expand_dims(roi,axis=0)

            preds=classifier.predict(roi)[0]

            label1=class_labels1[preds.argmax()]

            label_position1=(x,y)
```

```
cv2.putText(frame,label1,label_position1,cv2.FONT_HERSHEY_SIMPLEX,2,(0,255,0),3)
            label2=class_labels2[preds.argmax()]
            label_position2=(x,y-50)
cv2.putText(frame,label2,label_position2,cv2.FONT_HERSHEY_SIMPLEX,2,(0, 255, 255),3)
        else:
            cv2.putText(frame,'No Face Found',(20,20),cv2.FONT_HERSHEY_SIMPLEX,2,(0,255,0),3)


    cv2.imshow('Emotion Detector',frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
cap.release()
cv2.destroyAllWindows()
```

APP.JS

```
import MainScreen from
"./components/MainScreen/MainScreen.component";
import firepadRef, { db, userName } from "./server/firebase";
import "./App.css";
import { useEffect } from "react";
import {
```

```
  setMainStream,
  addParticipant,
  setUser,
  removeParticipant,
  updateParticipant,
} from "./store/actioncreator";
import { connect } from "react-redux";
function App(props) {
  const getUserStream = async () => {
    const localStream = await
navigator.mediaDevices.getUserMedia({
      audio: true,
      video: true,
    });
    return localStream;
  };
  useEffect(async () => {
    const stream = await getUserStream();
    stream.getVideoTracks()[0].enabled = false;
    props.setMainStream(stream);

    connectedRef.on("value", (snap) => {
```

```
      if (snap.val()) {
        const defaultPreference = {
          audio: true,
          video: false,
          screen: false,
        };
        const userStatusRef = participantRef.push({
          userName,
          preferences: defaultPreference,
        });
        props.setUser({
          [userStatusRef.key]: { name: userName,
...defaultPreference },
        });
        userStatusRef.onDisconnect().remove();
      }
    });
  }, []);
  const connectedRef = db.database().ref(".info/connected");
  const participantRef = firepadRef.child("participants");
  const isUserSet = !!props.user;
  const isStreamSet = !!props.stream;
```

```
  useEffect(() => {
    if (isStreamSet && isUserSet) {
      participantRef.on("child_added", (snap) => {
        const preferenceUpdateEvent = participantRef
          .child(snap.key)
          .child("preferences");
        preferenceUpdateEvent.on("child_changed",
(preferenceSnap) => {
          props.updateParticipant({
            [snap.key]: {
              [preferenceSnap.key]: preferenceSnap.val(),
            },
          });
        });
        const { userName: name, preferences = {} } =
snap.val();
        props.addParticipant({
          [snap.key]: {
            name,
            ...preferences,
          },
        });
```

```
      });

      participantRef.on("child_removed", (snap) => {

        props.removeParticipant(snap.key);

      });

    }

  }, [isStreamSet, isUserSet]);

  return (

    <div className="App">

      <MainScreen />

    </div>

  );

}

const mapStateToProps = (state) => {

  return {

    stream: state.mainStream,

    user: state.currentUser,

  };

};

const mapDispatchToProps = (dispatch) => {

  return {

    setMainStream: (stream) =>
dispatch(setMainStream(stream)),
```

```
    addParticipant: (user) => dispatch(addParticipant(user)),

    setUser: (user) => dispatch(setUser(user)),

    removeParticipant: (userId) =>
dispatch(removeParticipant(userId)),

    updateParticipant: (user) =>
dispatch(updateParticipant(user)),

  };

};

export default connect(mapStateToProps,
mapDispatchToProps)(App);
```

Peerconnection.js

```
import firepadRef from "./firebase";

import { store } from "../index";

const participantRef = firepadRef.child("participants");

export const updatePreference = (userId, preference) => {

  const currentParticipantRef = participantRef

    .child(userId)

    .child("preferences");

  setTimeout(() => {

    currentParticipantRef.update(preference);

  });

};
```

```
export const createOffer = async (peerConnection, receiverId,
createdID) => {

  const currentParticipantRef =
participantRef.child(receiverId);

  peerConnection.onicecandidate = (event) => {

    event.candidate &&

      currentParticipantRef

        .child("offerCandidates")

        .push({ ...event.candidate.toJSON(), userId: createdID
});

  };

  const offerDescription = await peerConnection.createOffer();

  await peerConnection.setLocalDescription(offerDescription);


  const offer = {

    sdp: offerDescription.sdp,

    type: offerDescription.type,

    userId: createdID,

  };

  await currentParticipantRef.child("offers").push().set({
offer });

};

export const initializeListensers = async (userId) => {
```

```
  const currentUserRef = participantRef.child(userId);

  currentUserRef.child("offers").on("child_added", async
(snapshot) => {

    const data = snapshot.val();

    if (data?.offer) {

      const pc =

store.getState().participants[data.offer.userId].peerConnectio
n;

      await pc.setRemoteDescription(new
RTCSessionDescription(data.offer));

      await createAnswer(data.offer.userId, userId);

    }

  });

  currentUserRef.child("offerCandidates").on("child_added",
(snapshot) => {

    const data = snapshot.val();

    if (data.userId) {

      const pc =
store.getState().participants[data.userId].peerConnection;

      pc.addIceCandidate(new RTCIceCandidate(data));

    }

  });
```

```
  currentUserRef.child("answers").on("child_added", (snapshot)
=> {
    const data = snapshot.val();
    if (data?.answer) {
      const pc =

store.getState().participants[data.answer.userId].peerConnecti
on;
      const answerDescription = new
RTCSessionDescription(data.answer);
      pc.setRemoteDescription(answerDescription);
    }
  });
  currentUserRef.child("answerCandidates").on("child_added",
(snapshot) => {
    const data = snapshot.val();
    if (data.userId) {
      const pc =
store.getState().participants[data.userId].peerConnection;
      pc.addIceCandidate(new RTCIceCandidate(data));
    }
  });
};
const createAnswer = async (otherUserId, userId) => {
```

```
  const pc =
store.getState().participants[otherUserId].peerConnection;

  const participantRef1 = participantRef.child(otherUserId);

  pc.onicecandidate = (event) => {

    event.candidate &&

      participantRef1

        .child("answerCandidates")

        .push({ ...event.candidate.toJSON(), userId: userId
});

  };

  const answerDescription = await pc.createAnswer();

  await pc.setLocalDescription(answerDescription);

  const answer = {

    type: answerDescription.type,

    sdp: answerDescription.sdp,

    userId: userId,

  };

  await participantRef1.child("answers").push().set({ answer
});

};
```

MISCELLANEOUS

To make sure we have a working directory for the application we have made, We have installed multiple packages like :

- For CNN -
- Tensorflow
- Keras
- OpenCV
- Matplotlib

- For Meet Application:
- NPM
- React-scripts
- Node.Js
- WebRTC

In this section we will be directing you as to how to install the packages.

- Tensorflow:

Install Tensorflow using pip package manager by running the command "pip install tensorflow".

You can also install a specific version of Tensorflow by specifying the version number after the package name. For example, "pip install tensorflow==2.4.0".

- Keras:

Keras is now part of the Tensorflow package, so it should be automatically installed when you install Tensorflow. If you need to install Keras separately, you can run the command "pip install keras".

- OpenCV:

Install OpenCV using pip package manager by running the command "pip install opencv-python".

You can also install additional modules for OpenCV by running the command "pip install opencv-contrib-python".

- Matplotlib:

Install Matplotlib using pip package manager by running the command "pip install matplotlib".

Matplotlib also has additional packages that can be installed for specific functionalities such as interactive plotting or basemaps. To install these packages, you can refer to the Matplotlib documentation.

Make sure you have the latest version of pip installed before installing these packages. You can do this by running the command "pip install --upgrade pip". It is also recommended to create a virtual environment to avoid any conflicts between packages installed for different projects.

NPM (Node Package Manager): NPM is used to manage packages and dependencies for Node.js applications. It is installed along with Node.js. To check if you have NPM installed, run the command npm -v in your terminal. If it is not installed, download and install Node.js from the official website, and NPM will be installed automatically.

React-scripts: React-scripts is a set of scripts and configurations used by Create React App (CRA) to create a new React project. To install React-scripts, first, make sure you have Node.js and NPM installed. Then, open your terminal and run the command npm install -g create-react-app to install the CRA package globally. Next, run the command create-react-app my-app to create a new React project. Finally, navigate to the project directory and run the command npm start to start the development server.

Node.js: Node.js is a JavaScript runtime built on Chrome's V8 JavaScript engine. To install Node.js, download the installer from the official website and run it on your machine. Follow the installation wizard's instructions to complete the installation process.

WebRTC: WebRTC (Web Real-Time Communication) is a set of APIs used for real-time communication in web browsers. To use WebRTC, you need to install a compatible browser, such as Google Chrome, Mozilla Firefox, or Microsoft Edge. There is no separate installation process required for WebRTC.

Note: The installation process may vary depending on your operating system and package manager. It is always recommended to refer to the official documentation for detailed installation instructions.

PLAGIARISM REPORT

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
(Deemed to be University u/s 3 of UGC Act, 1956)

Office of Controller of Examinations

REPORT FOR PLAGIARISM CHECK ON THE SYNOPSIS/THESIS/DISSERTA/PROJECT REPORTS

1	Name of the Candidate **(IN BLOCK LETTERS)**	**1. Om Rajput** **2. Suyash Nigam**
2	Address of the Candidate	1. Sahakar Nagar,Bangalore 2. Nehru Nagar, Kanpur **Mobile Number:** 1. 9008382917 2. 9838936135
3	Registration Number	1. RA1911032010028 2. RA1911032010027
4	Date of Birth	1. 30/11/2001 2. 23/02/2001
5	Department	Department of Networking and Communications
6	Faculty	Engineering and Technology
7	Title of the Synopsis/ Thesis/ Dissertation/Project	COMPUTER VISION-BASED WARNING SYSTEM OF LEARNERS ENGAGEMENT SHIFT IN ONLINE LEARNING PLATFORMS
8	Name and address of the Supervisor /Guide	Dr. Naga Malleswari TYJ Associate Professor SRM Institute of Science and Technology **Mail ID:** nagamalt@srmist.edu.in **Mobile Number:** 9176314249
9	Software Used	Turnitin
10	Date of Verification	09/05/2023

11	**Plagiarism Details: (to attach the final report)**			
Chapter	**Title of the Chapter**	**Percentage of similarity index (including self citation)**	**Percentage of similarity index (Excluding self citation)**	**% of plagiarism after excluding Quotes, Bibliograph y, etc.,**
1	Introduction	2%	2%	2%
2	Literature Review	3%	3%	3%
3	System Design	<1%	<1%	<1%
5	Proposed Work	1%	1%	1%
5	Implementation	1%	1%	1%
6	Testing	<1%	<1%	<1%
7	Results & Discussion	0%	0%	0%
8	Conclusion	<1%	<1%	<1%
9	Future Enhancement	0%	0%	0%

We declare that the above information has been verified and found true to the best of our knowledge.

Signature of the Candidate	**Signature of the Supervisor / Guide**
Signature of the Co-Supervisor/Co-Guide	**Signature of the HOD / DRCC Chairperson**

Signature of the Librarian

Report Final

ORIGINALITY REPORT

9%	5%	6%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	developers.google.com Internet Source	1%
2	www.researchgate.net Internet Source	1%
3	arxiv.org Internet Source	1%
4	"Handbook of Research on Student Engagement", Springer Nature, 2012 Publication	<1%
5	Submitted to Uttar Pradesh Technical University Student Paper	<1%
6	"Machine Learning and Big Data Analytics (Proceedings of International Conference on Machine Learning and Big Data Analytics (ICMLBDA) 2021)", Springer Science and Business Media LLC, 2022 Publication	<1%
7	Submitted to University of North Texas Student Paper	<1%

8	journals.plos.org Internet Source	<1 %
9	Submitted to Gitam University Student Paper	<1 %
10	Submitted to The University of Wolverhampton Student Paper	<1 %
11	link.springer.com Internet Source	<1 %
12	www.kdnuggets.com Internet Source	<1 %
13	www.mdpi.com Internet Source	<1 %
14	Chakradhar Pabba, Praveen Kumar. "An intelligent system for monitoring students' engagement in large classroom teaching through facial expression recognition", Expert Systems, 2021 Publication	<1 %
15	"Transforming Learning with Meaningful Technologies", Springer Science and Business Media LLC, 2019 Publication	<1 %
16	www.science.gov Internet Source	<1 %

17	"International Conference on Innovative Computing and Communications", Springer Science and Business Media LLC, 2022 Publication	<1 %
18	hackr.io Internet Source	<1 %
19	Sarah Deniz, Dakyung Lee, Grace Kurian, Lourdes Altamirano et al. "Computer Vision for Attendance and Emotion Analysis in School Settings", 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), 2019 Publication	<1 %
20	rcastoragev2.blob.core.windows.net Internet Source	<1 %
21	www.nature.com Internet Source	<1 %
22	"Machine Learning and Information Processing", Springer Science and Business Media LLC, 2021 Publication	<1 %
23	Felipe Zago Canal, Tobias Rossi Müller, Jhennifer Cristine Matias, Gustavo Gino Scotton et al. "A survey on facial emotion recognition techniques: A state-of-the-art literature review", Information Sciences, 2022 Publication	<1 %

24	Submitted to University of Edinburgh Student Paper	<1 %
25	Submitted to IUBH - Internationale Hochschule Bad Honnef-Bonn Student Paper	<1 %
26	Submitted to Palm Beach State College Student Paper	<1 %
27	J. Ejaz, M. A. Shah. "Real-time analysis of student's behavioural engagement in digital smart classrooms using fog computing and IoT devices", Competitive Advantage in the Digital Economy (CADE 2022), 2022 Publication	<1 %
28	M. Ali Akber Dewan, Mahbub Murshed, Fuhua Lin. "Engagement detection in online learning: a review", Smart Learning Environments, 2019 Publication	<1 %
29	Submitted to University of Sydney Student Paper	<1 %
30	Chen, X., Z. Jin, G. Lin, and X. Yang. "A novel stereo-model retrieval method in manufacturing", Proceedings of the Institution of Mechanical Engineers Part B Journal of Engineering Manufacture, 2015. Publication	<1 %
31	Joseph Yiu. "Technical Overview", Elsevier BV, 2015	<1 %

Publication

32	Mayanda Mega Santoni, T. Basaruddin, Kasiyah Junus. "Convolutional Neural Network Model based Students' Engagement Detection in Imbalanced DAiSEE Dataset", International Journal of Advanced Computer Science and Applications, 2023 Publication	<1 %
33	Submitted to Middlesex University Student Paper	<1 %
34	Submitted to Navitas Global (UK SITES) Student Paper	<1 %
35	Poonam Sharma, R. N. Yadav, K. V. Arya. "Face recognition from video using generalized mean deep learning neural network", 2016 4th International Symposium on Computational and Business Intelligence (ISCBI), 2016 Publication	<1 %
36	Submitted to Symbiosis International University Student Paper	<1 %
37	assets.researchsquare.com Internet Source	<1 %
38	kirenz.github.io Internet Source	<1 %

39	Aurobind V. Iyer, Viral Pasad, Smita R. Sankhe, Karan Prajapati. "Emotion based mood enhancing music recommendation", 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), 2017 Publication	<1 %
40	Pojala Chiranjeevi, Viswanath Gopalakrishnan, Pratibha Moogi. "Neutral Face Classification Using Personalized Appearance Models for Fast and Robust Emotion Detection", IEEE Transactions on Image Processing, 2015 Publication	<1 %
41	Submitted to Cerritos College Student Paper	<1 %
42	pinpdf.com Internet Source	<1 %
43	scholarworks.unist.ac.kr Internet Source	<1 %
44	www.akademisains.gov.my Internet Source	<1 %
45	www.globalbenchmarking.com Internet Source	<1 %
46	www.internationaljournalssrg.org Internet Source	<1 %

47	Feng Pan, Bingyao Huang, Chunhong Zhang, Xinning Zhu, Zhenyu Wu, Moyu Zhang, Yang Ji, Zhanfei Ma, Zhengchen Li. "A survival analysis based volatility and sparsity modeling network for student dropout prediction", PLOS ONE, 2022 Publication	<1%
48	Submitted to South Dakota Board of Regents Student Paper	<1%
49	Thusnavis Bella Mary I, P. Malin Bruntha, M. A. P. Manimekalai, K. Martin Sagayam, Hien Dang. "Investigation of an Efficient Integrated Semantic Interactive Algorithm for Image Retrieval", Pattern Recognition and Image Analysis, 2021 Publication	<1%
50	Submitted to Tower Hamlets College Student Paper	<1%
51	Submitted to Wesleyan University Student Paper	<1%
52	"Human-Computer Interaction – INTERACT 2021", Springer Science and Business Media LLC, 2021 Publication	<1%
53	Junping Hu, Shitu Abubakar, Shengjun Liu, Xiaobiao Dai, Gen Yang, Hao Sha. "Near-Infrared Road-Marking Detection Based on a	<1%

Modified Faster Regional Convolutional Neural Network", Journal of Sensors, 2019
Publication

---

54 Prakhar Bhardwaj, P.K. Gupta, Harsh Panwar, Mohammad Khubeb Siddiqui, Ruben Morales-Menendez, Anubha Bhaik. "Application of Deep Learning on Student Engagement in e-learning environments", Computers & Electrical Engineering, 2021 <1%
Publication

---

55 Rodrigo N. Calheiros, Rajiv Ranjan, Anton Beloglazov, César A. F. De Rose, Rajkumar Buyya. "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms", Software: Practice and Experience, 2011 <1%
Publication

---

56 Tzuu-Hseng S. Li, Ping-Huan Kuo, Ting-Nan Tsai, Po-Chien Luan. "CNN and LSTM Based Facial Expression Analysis Model for a Humanoid Robot", IEEE Access, 2019 <1%
Publication

---

57 maxwellsci.com <1%
Internet Source

---

58 openportal.isti.cnr.it <1%
Internet Source

59	ouci.dntb.gov.ua Internet Source	<1 %
60	repository.tudelft.nl Internet Source	<1 %
61	"Foundations of Augmented Cognition: Neuroergonomics and Operational Neuroscience", Springer Nature, 2016 Publication	<1 %
62	Sarra Ayouni, Fahima Hajjej, Mohamed Maddeh, Shaha Al-Otaibi. "A new ML-based approach to enhance student engagement in online environment", PLOS ONE, 2021 Publication	<1 %
63	Katherine Covell. "School engagement and rights ‐ respecting schools", Cambridge Journal of Education, 2010 Publication	<1 %
64	digitalcommons.njit.edu Internet Source	<1 %

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		

PUBLICATION PROOF

16 of 29

Confirmation of your submission to International Journal of Educational Technology in Higher Education - ETHE-D-23-00485 - [EMID:d1274323b9eba7de] Inbox ×

Int. Journal of Educational Technology in Higher Education Editorial Office <em@editorialmanager.com> Mon, Apr 17, 9:43 AM

to me

ETHE-D-23-00485
Computer vision based warning system of learners engagement shift in online learning platforms
Om Rajput; Suyash Nigam; Naga Malleswari TYJ; Ushasukhanya S
International Journal of Educational Technology in Higher Education

Dear Mr. Rajput,

Thank you for submitting your manuscript,
"Computer vision based warning system of learners engagement shift in online learning platforms", to International Journal of Educational Technology in Higher Education

The submission id is: ETHE-D-23-00485
Please refer to this number in any future correspondence.

During the review process, you can keep track of the status of your manuscript by accessing the following website:

https://www.editorialmanager.com/ethe/

If you have forgotten your username or password please use the "Send Login Details" link to get your login information. For security reasons, your password will be reset.

Best wishes,

Editorial Office
International Journal of Educational Technology in Higher Education