



VIT[®]
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

School of Information Technology & Engineering

Department of Computer Application

Fall Semester 2022-23

ITA6099 – Master Thesis

Third Review Report

Reg. No: **21MCA0153**

Student Name: **VIVEK KUMAR**

Guide Name: **Dr. VIJAYAN R**

Master Thesis Title:

**STUDENT BEHAVIOR AND ENGAGEMENT ANALYSIS IN ONLINE CLASS
USING DLIB HoG DETECTION**

Programming Languages/ Tools / DBMS Used for Implementation:
(Both Front End and Back End Details)

Programming Language: PYTHON (Backend)

Front End: JavaScript

DBMS Used: SQLite

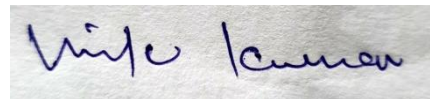
DECLARATION

I here by declare that the thesis entitled “**STUDENT BEHAVIOR AND ENGAGEMENT ANALYSIS IN ONLINE CLASS USING DLIB HoG DETECTION**” submitted by me, for the award of the degree of **MASTER OF COMPUTER APPLICATIONS** is a record of bonafide work carried out by me under the supervision of **Dr. VIJAYAN R.**

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place: **Vellore Institute of Technology, Vellore**

Date: 10-03-2023



Signature of the Candidate

CERTIFICATE

This is to certify that the thesis entitled “**STUDENT BEHAVIOR AND ENGAGEMENT ANALYSIS IN ONLINE CLASS USING DLIB HoG DETECTION**” submitted by **VIVEK KUMAR (21MCA0153)**, School of Information Technology & Engineering, Vellore Institute of Technology, Vellore for the award of the degree MCA(Master of Computer Applications) is a record of bonafide work carried out by him/her under my supervision.

The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university. The Project report fulfils the requirements and regulations of **VELLORE INSTITUTE OF TECHNOLOGY, VELLORE** and in my opinion meets the necessary standards for submission.

Signature of the Guide

Signature of the HoD

Internal Examiner

External Examiner

ACKNOWLEDGEMENT

It is my pleasure to express with a deep sense of gratitude to **Dr. VIJAYAN R, School of Information Technology and Engineering, Vellore Institute of Technology**, for his constant guidance, continual encouragement and understanding; more than all, he taught me patience in my endeavour. My association with him is not confined to academics only, but it is a great opportunity on my part to work with an intellectual and expert in the field of Computer Science.

I would like to express my gratitude to **Dr. G.Viswanathan** founder and chancellor, **Dr. Sekar Viswanathan** Vice President, **Mr. G.V.Selvam** Vice President, **Mr. Sankar Viswanathan** Vice President, **Dr. Rambabu Kodali** Vice Chancellor, **Dr. S.Narayanan** Pro-Vice Chancellor and **Dr. Sumathy S** Dean, School of Information Technology and Engineering, for providing me with an environment to work in and for his inspiration during the tenure of the course.

In a jubilant mood, I express ingeniously my whole-hearted thanks to, **Dr. Vanitha.M**, Head of the Department, Department of Computer Applications, School of Information Technology and Engineering, all teaching staff and members working as limbs of our university for their not-self-centred enthusiasm coupled with timely encouragements showered on me with zeal, which prompted the acquirement of the requisite knowledge to finalize my course study successfully. I would like to thank my parents for their support.

It is indeed a pleasure to thank my friends who persuaded and encouraged me to take up and complete this task. Last, but not least, I express my gratitude and appreciation to all those who have helped me directly or indirectly toward the successful completion of this project.

Place: **Vellore Institute of Technology, Vellore**

Date: 10-03-2023

Name of the student

VIVEK KUMAR (21MCA0153)

ABSTRACT:

Monitoring a student while they are in a video conferencing classroom is a non-intrusive approach to digitizing students' behaviour. A non-intrusive method of digitizing student behavior in video conferencing classrooms is through monitoring. Understanding the indicators of attention deficit is essential to improve the dynamics of a lecture. The purpose of our project is to develop an autonomous agent that provides information to both teachers and students. This innovation can enhance education by guiding teachers on improving their teaching approach and counseling students on how to improve their behavior and academic performance. Our project provides visual feedback to teachers about students' average attention level and counsels students regarding their conduct in class. For instance, we can analyze sessions where students were inattentive and identify topics that may require further attention. For example, I can sessions in which students were less watchful and the corresponding topics that potentially need extra attention.

Keywords: *Machine Learning, Face Recognition, OpenCV, dlib, HoG, Face Landmark estimation, SVM*

TABLE OF CONTENTS

DECLARATION.....	ii
CERTIFICATE	iii
ACKNOWLEDGEMENT	iv
ABSTRACT	v
LIST OF FIGURES	ix
LIST OF TABLES	xii
LIST OF ACRONYMS	xiii

INTRODUCTION

- 1.1 INTRODUCTION**
- 1.2 OVERVIEW OF THE PROJECT**
- 1.3 PROBLEM STATEMENT**
- 1.4 OBJECTIVES**
- 1.5 SCOPE OF THE PROJECT**

LITERATURE SURVEY

- 2.1 SUMMARY OF THE EXISTING WORKS**
- 2.2 CHALLENGES PRESENT IN EXISTING SYSTEM**

PROPOSED SYSTEM

- 3.1 IDEA OF THE PROPOSED SYSTEM**
- 3.2 ADVANTAGES OF PROPOSED WORK**

ANALYSIS & DESIGN

- 4.1 PROPOSED METHODOLOGY**
- 4.2 SYSTEM ARCHITECTURE**

MODULE DESCRIPTIONS

5.1 Dlib Detector

5.2 HoG-Histogram of Oriented Gradients

5.3 OpenCV (Open Source Computer Vision Library)

5.4 SVM (Support Vector Machine)

IMPLEMENTATION

6.1 SOFTWARE REQUIREMENTS

6.2 PROGRAMMING LANGUAGE USED

6.3 BASIC NEEDS

6.4 IMPLEMENTATION

6.5 RESULT ANALYSIS & EVALUATION METRICS

7. CONCLUSIONS AND FUTURE WORK

7.1 CONCLUSION

7.2 FUTURE WORK

8. REFERENCES

9. APPENDICES

LIST OF FIGURES

1.1 SYSTEM ARCHITECTURE

1.2 SYSTEM ARCHITECTURE FLOW

1.3 FACE LANDMARKS DETECTION

LIST OF TABLES

1.1 TASKS ACTIVITY

LIST OF ACRONYMS

WebRTC Web Real-Time Communications

HOG Histogram of Oriented Gradients

MULDA Multi-view Uncorrelated Linear Discriminant Analysis

INTRODUCTION

1.1 Introduction

Virtual learning environments have been widely adopted in the field of education, providing simulated interactions that are comparable to traditional teaching processes at a cognitive level. In the realm of human-computer interaction, facial recognition systems are expected to detect, analyze, and process emotions to enhance learning outcomes such as the perception, understanding, and expression of emotions. When students display various expressions in virtual learning environments, teachers can gauge their level of comprehension and tailor their teaching programs accordingly. This allows for more effective and personalized instruction, improving the overall quality of the learning experience.

Our project focuses on improving the quality of e-education in local college or university networks through one-to-many video communication, enhancing it with technologies such as facial detection using deep learning, real-time feedback using charts and alert systems, and more. We aim to make the learning experience more analysis-oriented by providing summaries at the end of video lectures for both teachers and students. By using an ML model to analyze data collected during the session, we can show the teacher the general attentiveness of students, highlighting durations where students felt disengaged or had doubts. This analysis provides the teacher with feedback to improve their course. Additionally, students receive feedback on their concentration during the lecture. Our project also includes a dashboard where both students and teachers can access their past performance and create a to-do list to finish tasks for a given day.

Emotional data of students in virtual environments can be obtained from video frames, but balancing accuracy and time is crucial for real-time understanding of emotions. Increasing accuracy requires more video frames, resulting in longer computation time. Conversely, decreasing precision means collecting fewer feature data for efficiency. Current solutions, such as Decision trees to identify facial expressions, have limitations that affect accuracy and efficiency. Therefore, optimizing emotion recognition in a virtual learning environment must balance both. This optimization can enable teachers to adjust teaching strategies and provide real-time feedback to students, ultimately improving teaching quality. To find the best solution for emotion recognition based on face recognition in a real-time virtual learning environment, accuracy and efficiency must be improved.

1.2 OVERVIEW OF THE PROJECT

The main objective of our project is to enhance the quality of e-education through the use of technology, specifically facial detection using deep learning, real-time feedback using charts and alert systems, and optimizing emotion recognition in a virtual learning environment. Our project aims to provide an analysis-oriented learning experience by analyzing data collected during video lectures and providing feedback to both teachers and students. The ML model's data is used to show the general attentiveness of all the students in the class to the teacher, highlighting periods where students were most disengaged or had the most doubts, providing the lecturer with an analysis of his/her performance and assisting in further course

improvement. The project includes a dashboard where both students and teachers can access their past performance and create a to-do list to finish tasks for a given day.

Overall, our project aims to improve the quality of e-education and provide a personalized learning experience for students. Through the use of facial detection and emotion recognition technology, we aim to provide teachers with the necessary tools to adjust their teaching strategies and provide students with the necessary feedback to improve their academic performance.

1.3 PROBLEM STATEMENT

- The current online education system lacks the ability to provide personalized feedback to both teachers and students, resulting in a one-size-fits-all approach that may not be effective for all learners.
- The absence of emotion recognition technology in virtual learning environments poses a significant challenge in improving the quality of e-education. Current emotion recognition solutions are either inaccurate or inefficient, making it challenging to provide real-time feedback to both teachers and students.
- Inefficient use of technology in e-education results in a lack of data analysis, making it difficult for teachers to adjust their teaching strategies and provide students with the necessary feedback to improve their academic performance.
- The absence of tools to analyze student behavior and engagement during virtual learning environments makes it challenging for teachers to identify students' specific learning needs and provide targeted feedback.
- Ineffective use of technology in e-education leads to a lack of engagement and disinterest among students, reducing the overall effectiveness of virtual learning environments.

1.4 OBJECTIVE OF PROJECT:

The aim of this project is to create an automated agent that can provide valuable insights to both teachers and students. It is well-established that students who are engaged tend to perform better academically and develop critical thinking skills. One of the ways to determine student engagement is through features and pose extraction, where head position plays a critical role in determining gaze direction. Although eye-tracking can also be used, it suffers from low resolution in images. By combining head position with another technique, the accuracy of attention detection can be vastly improved. In general, students who are paying attention tend to react similarly to stimuli, such as synchronizing their movements with the majority of the class. For instance, when the teacher instructs the class to bend down to write, attentive students would follow suit. Students that are paying attention normally react to stimulus the

same way, that is students having their motions synchronised to the majority are paying attention. An example of this synchronization is when the class has to bend down to write when the teachers instructs them.

1.5 SCOPE OF PROJECT:

The project has the potential to significantly impact education by offering guidance and feedback to teachers and students. By utilizing technology to monitor and analyze student behavior, the project can provide valuable insights to improve teaching strategies and enhance academic performance. This can ultimately lead to a more tailored and effective learning experience for all students.

By using technology to monitor and analyze student behavior in real-time, the project can provide valuable insights and feedback to both teachers and students. For teachers, the information can be used to adjust their teaching strategies and improve the effectiveness of their lessons. For students, the feedback can be used to improve their behavior and academic performance. This can lead to a more personalized and effective learning experience for all students.

LITERATURE SURVEY

2.1 Summary of the Existing Works:

I have intended on reviewing those research works and their system frameworks to analyze the existing research in the field of emotion recognition in a classroom environment. Specifically, I have selected four previous research works that focus on the analysis of student emotions in a classroom setting.

The selection of these papers were done based on the following criteria:

To survey the identification of facial emotions of students in a classroom, I have focused primarily on recent research papers that employ different methodologies for feature extraction and classification of emotions, as well as the hardware setup used in facial image processing. Two proposals have been examined for gauging student emotions in the classroom setting without direct supervision. Various approaches can be used for the detection and recognition of human faces, such as the Viola-Jones algorithm and appearance-based techniques, including Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), Independent Component Analysis (ICA), and Support Vector Machine (SVM). The use of feature descriptors, such as Histograms of Oriented Gradients (HOG), Gabor, and Local Binary Pattern (LBP), have also been compared in different studies. Hybrid techniques have been proposed to

improve facial recognition accuracy, such as combining HOG and SVM or using Convolutional Neural Networks (CNN), but CNN can be computationally expensive.

The proposed method for the classroom setting includes two stages. In stage one, a face detector is used to pinpoint multiple faces in images of various sizes and detect faces even in interlaced scenarios. In stage two, a CNN model is used for face classification as masked or non-masked.

Some research papers have focused on assessing classroom teaching based on student emotions, such as capturing the video of the teacher to predict his/her emotions along with the student's emotions. Other studies have proposed systems that track and analyze student behavior based on computer vision, including person detection and validation, facial feature extraction, gaze tracking, person counting and phone detection, and emotion recognition.

Several methods for cross-view classification have also been proposed, such as Canonical Correlation Analysis (CCA), Generalized Multiview Analysis (GMA), and Multi-view Uncorrelated Linear Discriminant Analysis (MULDA), with varying degrees of supervised or unsupervised learning. Other studies have employed facial expression methods, including eye gazes, mouse behavior, and SVM classifiers, to achieve high accuracy rates in identifying engagement levels. A proposed method for the identification of facial emotions of students in a classroom includes implementing a CNN model with transfer learning, which uses deep CNNs for image data.

[1] I have considered proposal on Mood Extraction Using Facial Features to Improve Learning Curves of Students in E-Learning Systems, where there is no direct supervision involved in gauging the emotions of the students.

[2] Several approaches can be used for the detection and recognition of human faces. Among the most well-known techniques, it can be mentioned the Viola-Jones algorithm, widely used for its efficiency in computing time, which allowed its application in real-time.

[3] There are also appearance-based techniques, such as Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), Independent Component Analysis (ICA), and Support Vector Machine (SVM), which use machine learning, to learn the characteristics of the model.

[4] Although the appearance-based approach has good results, it is relatively complex to implement, due to the need of many samples in the training stage. Feature descriptors are also applied as methods for face recognition. For instance makes a comparison between three famous facial feature descriptors, Histograms of Oriented Gradients (HOG), Gabor and Local Binary Pattern (LBP), where it presents the advantages and disadvantages of using each method. Considering embedded implementations, it can be also noted different techniques applied during the recognition phase.

[5] As the systems proposed in the same way as the work , some authors also propose the use of hybrid techniques to increase the accuracy of facial recognition. The authors in, uses the SVM technique combined with the HOG and Gabor filters to detect emotions in the human face, this demonstrates an improvement in the classification accuracy. The conclusion is that the solution of the combination of HOG and SVM presents the best average of the results.

[6] In, also tests the combination of HOG and SVM compared to the PCA for face recognition. This work shows that there is a higher precision of the first method in relation to the second. Other authors, such as suggest the use of more advanced features, such as the use of Convolutional Neural Network (CNN) for facial detection. The advantage of applying this type of net is the ease of adaptation when the face is in position or angle variations.

[7] However, the authors claim that CNN has a high computational cost and presents better performance when running on Graphics Processing Units a comparative analysis of the processing time for face detection is presented, between the hybrid method HOG + SVM, when compared to the use of CNN from Dlib.

[8] There are two major stages to it. A Face Detector is included in the stage 1 of our architecture, which pinpoints multiple faces in images of various sizes and detects faces even in interlaced scenarios. The detected faces (roi) are then batched together and passed to the stage 2 of our architecture, a CNN model for Face Classification as masked or non masked.

[9] They used Uniform Local Binary Patterns (Uniform LBP) with 8 neighbours and radius 1 to extract the appearance features, principal component analysis (PCA) to reduce the dimensionality of the descriptor, and Support Vector Machines with Radial Basis Function kernels to classify the data. The method used is generally that described for the static LBP method.

[10] They have proposed a local representation by using a residual curvature to analyse the distance variation of facial landmark action unit.. Secondly, to handle variations in facial expressions, we have integrated facial action coding system using DCNN. Lastly, the integrated radial curvature traced facial action coding system is developed to recover only emotions using statistical shape models.

[12] Another work, I have taken for our review was originally presented by Sahla K. S. and T. Senthil Kumar for assessing the classroom teaching based on student emotions. This work is unique in the way that they had designed the system to capture the video of the teacher to predict his/her emotions along with the student's emotions, hence providing a two-way dissection of the emotions experienced by both the instructor and the students, offering more insight into the classroom lectures.

[13] proposes a system which performs student behavioral tracking and analysis based on computer vision in this paper. The proposed method for the system includes 5 major parameters which are considered as input to determine the engagement and behavior of the student and are as follows: Person Detection and Validation, Facial Feature Extraction, Gaze Tracking, Person Counting and Phone detection, Emotion Recognition. By combining these and performing these in parallel by multithreading.

[14] MvSL based approaches for cross-view classification as below. These include Canonical Correlation Analysis (CCA), Generalized Multiview Analysis (GMA) , Multi-view Uncorrelated Linear Discriminant Analysis (MULDA), where the first one is a well-known unsupervised MvSL based method, the other two are supervised methods that take into consideration intra-view discriminancy.

[15] Whitehill et al.'s approach used the facial expression method with boost (Box filter) and SVM (Gabor method). They achieved an accuracy rate of 72.9% with four engagement levels.

Li et al.'s approach employed facial expressions like eye gazes, the mouse behavior method with its geometric features, and CLM and SVM classifiers. They used three engagement levels.

[16] The proposed method is to implement a CNN model with transfer learning. For image data, so called deep CNNs have proved to perform similar or even better than humans in some recognition tasks. One of the main assumptions of most CNN architectures is that the inputs to these networks are raw images; this aid them in encoding specific properties into the network structure. The CNN transforms the dimensions of the image layer by layer while the various activation functions in the different layers, until it finally reduces to a single vector of scores assigned to each class. These scores are arranged along the depth of the final layer.

[17] which is reminiscent of the “baseline method” that uses a static Uniform LBP, utilizes a dynamic dense area-based appearance descriptors and statistical machine learning techniques. We applied histograms of oriented gradients (HoG) to extract the appearance features by accumulating the gradient magnitudes for a set of orientations in 1-D histograms defined over a size-adaptive dense grid, and Support Vector Machines with Radial Basis Function kernels as base learners of emotions.

2.2 CHALLENGES PRESENT IN EXISTING SYSTEM

Challenges present in the existing system for identifying facial emotions of students in a classroom include:

- **Accuracy and Efficiency:** Balancing accuracy and efficiency is crucial for real-time understanding of emotions. Several approaches have been proposed, including appearance-based techniques, feature descriptors, and hybrid techniques, but they can be computationally expensive.
- **Hardware Limitations:** The hardware setup used in facial image processing can significantly impact the accuracy and efficiency of emotion recognition. The use of multiple cameras, lighting conditions, and image resolution can affect the quality of facial image processing.
- **Dataset Diversity:** The performance of emotion recognition models heavily depends on the diversity of the dataset used for training. Emotions are subjective and culturally dependent, making it challenging to gather a diverse dataset that represents a broad range of emotions.
- **Cross-view Classification:** The performance of emotion recognition models can be affected by the viewpoint of the camera. Cross-view classification methods have been proposed to address this challenge, but they require significant supervised or unsupervised learning.
- **Engaging Teacher Behavior:** Some research papers have focused on assessing classroom teaching based on student emotions, such as capturing the video of the teacher to predict his/her emotions along with the student's emotions. However, the accuracy of such models heavily relies on the teacher's behavior and facial expressions, which can be difficult to control and standardize.

Overall, improving the accuracy, efficiency, and dataset diversity of emotion recognition models remains a significant challenge in the existing system for identifying facial emotions of students in a classroom.

PROPOSED SYSTEM

3.1 IDEA OF THE PROPOSED SYSTEM

The idea of the proposed work is to develop an autonomous agent that can monitor students' behavior in video conferencing classrooms to enhance education.

The proposed work includes:

- **Non-Intrusive Monitoring:** The project aims to monitor students' behavior in video conferencing classrooms in a non-intrusive manner. This can enable teachers to understand the indicators of attention deficit and improve the dynamics of the lecture.
- **Autonomous Agent:** The proposed work aims to develop an autonomous agent that can provide information to both teachers and students. This can enhance education by guiding teachers on improving their teaching approach and counseling students on how to improve their behavior and academic performance.
- **Visual Feedback:** The project provides visual feedback to teachers about students' average attention level. This can enable teachers to analyze sessions where students were inattentive and identify topics that may require further attention.
- **Counseling Students:** The proposed work can counsel students regarding their conduct in class. This can help students understand the areas where they need to improve and take appropriate actions.
- **Topic Analysis:** The proposed work can analyze sessions in which students were less watchful and identify the corresponding topics that potentially need extra attention. This can enable teachers to improve their teaching approach and help students understand the topics better.

Overall, the proposed work aims to improve the quality of education by monitoring students' behavior in video conferencing classrooms and providing feedback to both teachers and students.

3.2 ADVANTAGES OF PROPOSED WORK

Based on the provided conclusion, here are some potential advantages of the proposed work:

- **Enhanced E-education Quality:** The project aims to enhance the quality of E-education in a local network by providing one-to-many video communication, live classes by teachers to students in a college or university network. This can lead to an improved learning experience for students, with increased access to high-quality educational content.
- **Analysis-Oriented Teaching:** By incorporating technologies such as Facial Detection using Deep Learning, real-time Feedback using Charts, and Alert systems, the project can help make teaching more analysis-oriented. This can enable teachers to gain insights into student engagement levels and identify areas where students may need additional support or guidance.
- **Improved Teacher Performance:** At the end of each lecture, the ML Model's data can be used to provide teachers with an analysis of their performance, highlighting areas where students were most bored or had the most doubts. This feedback can help teachers improve their teaching methods, making them more effective at engaging students and improving overall learning outcomes.
- **Automated Lecture Summaries:** The project can generate automated summaries of video lectures that can be accessed by both teachers and students. This can save time and effort for teachers who would otherwise have to manually summarize their lectures and can enable students to review the content more easily.
- **Real-Time Feedback:** The project incorporates real-time Feedback using Charts, allowing teachers to gain insights into student engagement levels during the lecture. This can enable teachers to adjust their teaching methods in real-time to better engage students and improve learning outcomes.
- **ML Model Analysis:** The project uses an ML model to analyze the attentiveness of students during the lecture, identifying periods where students were most bored or had the most doubts. This feedback can enable teachers to improve their teaching methods, making them more effective at engaging students and improving learning outcomes.
- **Alert Systems:** The project incorporates Alert systems that can generate alerts to teachers when students are not engaging or participating in the lecture. This can enable teachers to intervene in real-time, offering additional support or guidance to students who may be struggling.

Overall, the proposed work can help to enhance the quality of E-education, make teaching more analysis-oriented, and improve teacher performance, leading to improved learning outcomes for students.

ANALYSIS & DESIGN

4.1 PROPOSED METHODOLOGY:

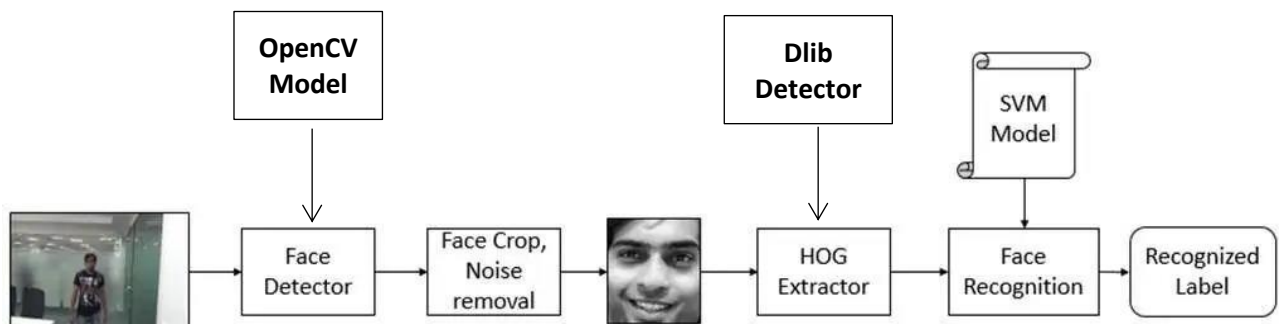
The proposed work aims to identify the most effective solution for emotion recognition using computer vision techniques. After conducting extensive research and experimentation, it has been determined that the combination of OpenCV and dlib libraries based on the HOG method yields the best average results. To extract appearance features, histograms of oriented gradients (HoG) were utilized by accumulating gradient magnitudes for a set of orientations in 1-D histograms defined over a size-adaptive dense grid. To classify emotions, Support Vector Machines with Radial Basis Function kernels were used as the base learners.

4.2 System Architecture:

Proposed project architecture is based on the use of the HOG (Histogram of Oriented Gradients) feature descriptor along with a linear SVM machine learning algorithm for face detection. HOG is a powerful and simple feature descriptor that is not only useful for face detection but also for object detection, such as cars, pets, and fruits. The HOG descriptor is robust for object detection because it characterizes the shape of the object using the local intensity gradient distribution and edge direction.

The basic idea of HOG is to divide the image into small connected cells, compute a histogram for each cell, and combine all the histograms to form a feature vector. This feature vector is unique for each face and is used for face detection.

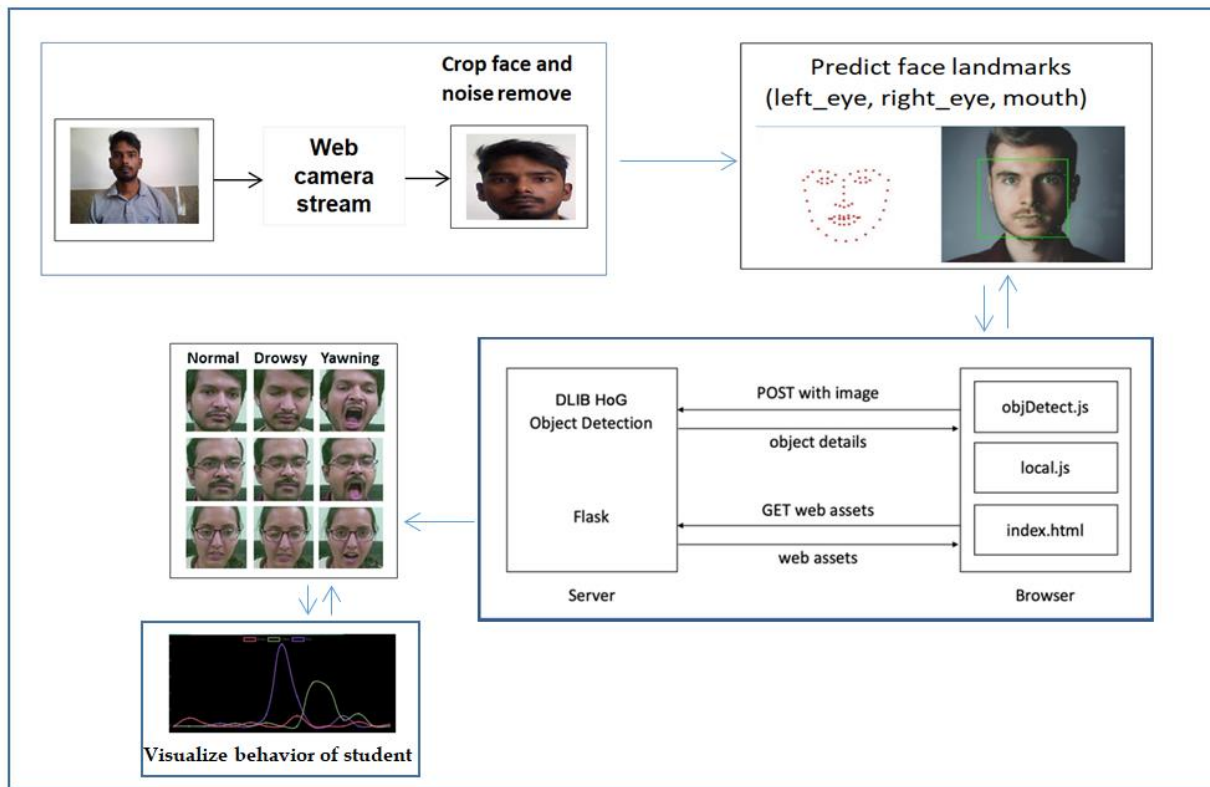
- The basic idea of HOG is dividing the image into small connected cells.
- Computes histogram for each cell
- Bring all histograms together to form feature vector i.e., it forms one histogram from all small histograms which is unique for each face



1.1 System Architecture

The system is designed to capture facial landmarks for object detection and perform computer vision (CV)-based tasks. The system will run multiple CV-based techniques in parallel and average their output to determine user engagement.

The techniques employed include:



1.2 System Architecture Flow

- Capture video stream from system camera.
- Convert video stream into individual frames.
- Extract 128-dimensional face embeddings.
- Detect faces using HoG and Linear SVM.
- Capture facial landmarks (mouth, head, and eyes) for detecting mouth movements.

MODULE DESCRIPTIONS

5.1 Dlib Detector:

Dlib Detector is a software library designed for machine learning and computer vision applications, developed by Davis King at the University of Maryland. The library provides a wide range of tools for developing complex applications in these fields, including image processing, face detection, object tracking, and machine learning algorithms.

One of the most popular features of Dlib Detector is its ability to detect faces in images and videos. It uses a combination of machine learning algorithms and image processing techniques to detect facial landmarks and extract features from images. The library provides a pre-trained model that can be used to detect faces in real-time video streams, making it a popular choice for facial recognition and tracking applications.

Dlib Detector is written in C++ and has bindings for several programming languages, including Python, Java, and C#. It is an open-source library and is freely available for commercial and non-commercial use. The library has been widely adopted in the computer vision and machine learning communities and is known for its robustness and efficiency.

Dlib Detector refers to the face detection algorithm provided by the Dlib library. This algorithm is based on the Histogram of Oriented Gradients (HOG) feature descriptor with a linear Support Vector Machine (SVM) machine learning algorithm. The Dlib Detector is widely used for face detection in various computer vision applications. The algorithm works by first dividing the image into small connected cells and computing histograms for each cell. These histograms are then combined to form a feature vector that is unique for each face. The SVM is trained on a large dataset of positive and negative face images to learn the discriminative features of faces. During face detection, the SVM is used to classify each image patch as either a face or non-face based on the learned features. If a face is detected, the algorithm returns the bounding box coordinates of the face in the image.

Dlib detector is an object detection algorithm based on a sliding window approach that uses the Histogram of Oriented Gradients (HoG) features to detect objects in an image. The basic idea behind the HoG feature descriptor has already been explained in a previous response.

Here is a brief overview of the Dlib detector with a diagram:

- Input image: The input image is first loaded into the algorithm.
- Sliding window: A sliding window of a fixed size is moved across the image. At each position, the content of the window is fed into the next step of the algorithm.
- Feature extraction: For each window, the HoG features are extracted from the image content. These features represent the gradient information of the pixels in the window.
- Classification: The HoG features are then fed into a classifier, which determines whether or not the window contains an object of interest. In Dlib detector, Support Vector Machine (SVM) is used as the classifier.
- Non-Maximum Suppression: The algorithm then applies a non-maximum suppression technique to remove overlapping detections and keep only the most confident detection.

- Output: The final output of the algorithm is the bounding box of the detected object in the image.

The Dlib detector is widely used for object detection tasks such as face detection, pedestrian detection, and object tracking.

5.2 HoG-Histogram of Oriented Gradients

Histogram of Oriented Gradients (HoG) is a popular feature descriptor used in computer vision and image processing for object detection and recognition. It was first introduced by Navneet Dalal and Bill Triggs in their paper "Histograms of Oriented Gradients for Human Detection" in 2005. The HoG feature descriptor works by capturing local gradients of an image. It divides an image into small rectangular regions called cells and computes a histogram of oriented gradients for each cell. The gradient is computed using the difference between the intensity values of adjacent pixels in the x and y direction. The orientation of the gradient is quantized into a number of bins, and the magnitude of the gradient is added to the corresponding bin in the histogram.

After computing the histograms for each cell, they are grouped into larger regions called blocks, and the histograms are normalized. Finally, the normalized block histograms are concatenated to form the HoG feature vector that characterizes the appearance of the object in the image. HoG is a robust feature descriptor that is invariant to changes in illumination, contrast, and scale. It is widely used in various computer vision applications, such as pedestrian detection, face detection, and object recognition.

Histogram of Oriented Gradients (HoG) is a feature descriptor used for object detection in computer vision.

The algorithm computes a histogram of gradient directions for each cell in a dense grid covering the image, and concatenates these histograms to form a feature vector.

Here is the mathematical formula to compute HoG:

- Compute gradient image $G(x, y)$ for the input image $I(x, y)$
- Divide the image into small connected cells. For each cell, compute gradient magnitude and orientation: $G_{mag}(x, y) = \sqrt{G_x(x, y)^2 + G_y(x, y)^2}$ $G_{dir}(x, y) = \text{atan2}(G_y(x, y), G_x(x, y))$ where G_x and G_y are the x and y derivatives of $G(x, y)$
- Divide the gradient orientations into K bins. Typically, $K = 9$. For each cell, create a histogram of gradient orientations weighted by the gradient magnitudes. Each histogram has K bins representing the gradient directions. The weight of each gradient is proportional to its magnitude. The orientation of the gradient is quantized into one of the K bins.
- Optionally, apply normalization to the histograms within a block. A block is a group of adjacent cells. Normalization helps to reduce the effects of illumination variations.
- Concatenate the histograms from all cells in the image to form the final feature vector.

5.3 OpenCV:

OpenCV (Open Source Computer Vision) is a powerful open-source library designed for real-time computer vision and image processing applications. It provides a wide range of image processing and computer vision algorithms that can be used for various applications such as object detection, face recognition, augmented reality, and robotics.

Python OpenCV is a Python library that provides a Python interface for the OpenCV library. This module allows users to access all the functions and tools provided by OpenCV from within Python code.

Python OpenCV is widely used in research, academia, and industry for computer vision applications. Python OpenCV provides a set of functions and tools for image and video processing, including image manipulation, filtering, and transformation, feature extraction, object detection, and machine learning. It also provides support for reading and writing image and video files, as well as streaming video from webcams and other sources. Python OpenCV is built on top of the C++ OpenCV library and provides a high-level Python interface that is easy to use and understand. It includes a range of utility functions and classes that make it easy to work with images and video data.

Some of the most commonly used classes in Python OpenCV include:

- `cv2.VideoCapture`: This class is used to capture video from a camera or a file.
- `cv2.imshow`: This function is used to display an image or a video on the screen.
- `cv2.imread`: This function is used to read an image from a file.
- `cv2.imwrite`: This function is used to save an image to a file.
- `cv2.cvtColor`: This function is used to convert an image from one color space to another.
- `cv2.CascadeClassifier`: This class is used for object detection using Haar cascades.
- `cv2.Tracker`: This class is used for object tracking.

Python OpenCV also includes support for machine learning algorithms such as k-nearest neighbors (k-NN), support vector machines (SVMs), and deep learning algorithms such as convolutional neural networks (CNNs). These algorithms can be used for tasks such as image classification, object recognition, and face detection. Python OpenCV can be installed using the pip package manager, and it is available for Windows, Linux, and macOS. It is compatible with Python 2 and 3, making it accessible to a wide range of users.

In summary, Python OpenCV is a powerful library that provides a range of functions and tools for image and video processing, object detection, feature extraction, and machine learning. It is

widely used in research, academia, and industry for computer vision applications, and it is easy to use and understand, making it accessible to users with different levels of expertise in computer vision and image processing.

GANTT CHART:



Acitivity	Description of the Activity	Guide Remarks
1		
2		
3		

1.1 TASK ACTIVITY

IMPLEMENTATION

6.1 HARDWARE REQUIREMENTS

- Graphics(GPU) 1GB
- Processing(CPU) i5+
- RAM Up to 4GB (2666 MHz)
- Storage Up to 50GB SSD
- Display 15.1"
- Camera 2MP+

6.2 SOFTWARE REQUIREMENTS

- Visual C++ Redistributable
- Dlib Library
- Flask Library

6.3 PROGRAMMING LANGUAGE USED

- Python 3.8+
- JavaScript

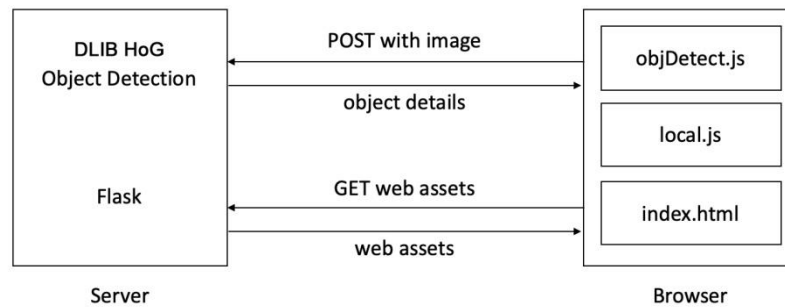
6.4 BASIC NEEDS

- Visual C++ Redistributable
- Flask Framework
- Camera / Microphone

6.5 Implementation

1. API Server :

To build a simple real-time object detection system, we can start with a basic architecture that streams video from a local web camera using WebRTC's `getUserMedia`, sends it to a Python server using the Flask web server, and applies the DLIB HoG Object Detection API for object detection. The architecture can be visualized as shown in the diagram below.



1.3 Face landmarks detection

The Flask web server will handle the serving of HTML and JavaScript files to the browser for rendering. Using the `getUserMedia.js` library, we can grab the local video stream. We will then use `objDetect.js` to send the images to the DLIB HoG Object Detection API via HTTP POST method. This will return the objects detected in the images along with their locations. We will package this information into a JSON object and send it back to `objDetect.js`, where we will display boxes and labels identifying the detected objects.

Object Detection Server generate four objects:

1. **classes** – an array of object names
2. **scores** – an array of confidence scores
3. **boxes** – locations of each object detected
4. **num** – the total number of objects detected

The arrays `classes`, `scores`, and `boxes` are of equal size and parallel to each other. Specifically, `classes[n]` corresponds to `scores[n]` and `boxes[n]`.

We need to set a threshold value for the scores generated by TensorFlow. By default, TensorFlow returns up to 100 objects, but many of these objects may be nested or overlapping with higher confidence objects. There are no established best practices for selecting a threshold value, but a value of 50% seems to work well for the sample images.

Sample Codes:

```
import os

from flask import Flask, request, Response, jsonify, render_template

import cv2

from FaceAction import FaceAction

from PIL import Image

import numpy
```



```

import time

app = Flask(__name__)

mydict = {}

rooms = {}

@app.after_request
def after_request(response):

    response.headers.add('Access-Control-Allow-Credentials','false'),

    response.headers.add('Access-Control-Allow-Origin', '*'),

    response.headers.add('Access-Control-Allow-Headers',

                          'Content-Type,Authorization')

    response.headers.add('Access-Control-Allow-Methods',
'GET,PUT,POST,DELETE,OPTIONS')

    return response


@app.route('/')
def index():

    return Response(open('./static/local.html').read(), mimetype="text/html")


def last5secAverage(prevc, newc, prevavg, newavg):

    return (newavg*newc-prevavg*prevc)/(newc-prevc)


@app.route('/image', methods=['POST', 'OPTIONS'])
def image():

    image_file = request.files['image']

    name = request.form['name']

    room = request.form['room']

    docopen = request.form['docopen']

    teacher = request.form['teacher']

    end = request.form['end']

```

```

print(end)

image_object = numpy.array(Image.open(image_file).convert('RGB'))

image_object = image_object[:, :, ::-1].copy()

drow, yawn, pos, number = FaceAction().run_frame(image_object)

drow_val = drow

if (drow < 0.2):
    drow = 1
else:
    drow = 0

if (yawn > 0.3):
    yawn = 1
else:
    yawn = 0

if(docopen == "false"):
    docopen = 0
else:
    docopen = 1

# print(docopen)

if room in rooms:
    if name in rooms[room]:
        if (end == '1'):
            rooms[room]['class&']['ClassEndTime'] = time.time()
            #print("I am here")
            rooms[room][name]['drow'] = drow
            rooms[room][name]['yawn'] = yawn
            rooms[room][name]['pos'] = pos
            rooms[room][name]['number'] = number
            rooms[room][name]['docopen'] = docopen
            if (rooms[room][name]['drow_val'] == drow_val):
                rooms[room][name]['paused'] = 1
            else:
                rooms[room][name]['paused'] = 0

```

```

rooms[room][name]['drow_val'] = drow_val

rooms[room][name]['avgdrow'] = (rooms[room][name]['avgdrow'] *
                                rooms[room][name]['count']+rooms[room][name]['drow']) /\
    (rooms[room][name]['count'] + 1)

rooms[room][name]['avgyawn'] = (rooms[room][name]['avgyawn'] *
                                rooms[room][name]['count']+rooms[room][name]['yawn']) /\
    (rooms[room][name]['count'] + 1)

rooms[room][name]['avgpos'] = (rooms[room][name]['avgpos'] *
                                rooms[room][name]['count']+rooms[room][name]['pos']) /\
    (rooms[room][name]['count'] + 1)

rooms[room][name]['avgdocopen'] = (rooms[room][name]['avgdocopen'] *
                                    rooms[room][name]['count']+rooms[room][name]['docopen']) /\
    (rooms[room][name]['count']+1)

rooms[room][name]['count'] += 1

# Dont update if Not going in if condition
rooms[room][name]['update'] = 0

nowtime = time.time()

#print((nowtime - rooms[room][name]['last5']))

if ((nowtime - rooms[room][name]['last5']) >= 5):
    # print("I am here")

    rooms[room][name]['lastavgdrow'] = last5secAverage(
        rooms[room][name]['pcount'],                      rooms[room][name]['count'],
rooms[room][name]['pavgdrow'], rooms[room][name]['avgdrow'])

    rooms[room][name]['lastavgyawn'] = last5secAverage(
        rooms[room][name]['pcount'],                      rooms[room][name]['count'],
rooms[room][name]['pavgyawn'], rooms[room][name]['avgyawn'])

    rooms[room][name]['lastavgpos'] = last5secAverage(
        rooms[room][name]['pcount'],                      rooms[room][name]['count'],
rooms[room][name]['pavgpos'], rooms[room][name]['avgpos'])

    rooms[room][name]['lastavgdocopen'] = last5secAverage(
        rooms[room][name]['pcount'],                      rooms[room][name]['count'],
rooms[room][name]['pavgdocopen'], rooms[room][name]['avgdocopen'])

```

```

rooms[room][name]['update'] = 1 # Update Graph if here
rooms[room][name]['drow_val'] = drow_val
rooms[room][name]['last5'] = nowtime
rooms[room][name]['pavgdrow'] = rooms[room][name]['avgdrow']
rooms[room][name]['pavgyaw'] = rooms[room][name]['avgyaw']
rooms[room][name]['pavgpos'] = rooms[room][name]['avgpos']
rooms[room][name]['pavgdocopen'] = rooms[room][name]['avgdocopen']
rooms[room][name]['pcount'] = rooms[room][name]['count']

```

We have to update Class Avg only when req is coming from teacher

```

if (teacher == "true"):

```

```

    avg_drow = 0

```

```

    avg_yaw = 0

```

```

    avg_pos = 0

```

```

    avg_docopen = 0

```

```

    ccc = 0

```

```

    for x in rooms[room]:

```

```

        if (x != 'class&'):

```

```

            # print(x)

```

```

            # print(rooms[room][x]['lastavgyaw'])

```

```

            avg_drow += rooms[room][x]['lastavgdrow']

```

```

            avg_yaw += rooms[room][x]['lastavgyaw']

```

```

            avg_pos += rooms[room][x]['lastavgpos']

```

```

            avg_docopen += rooms[room][x]['lastavgdocopen']

```

```

            ccc += 1

```

```

rooms[room]['class&']['Cdrow'] = avg_drow / ccc

```

```

rooms[room]['class&']['Cyawn'] = avg_yaw / ccc

```

```

rooms[room]['class&']['Cpos'] = avg_pos / ccc

```

```

rooms[room]['class&']['Cdocopen'] = avg_docopen / ccc

```

```

else:

```

```

rooms[room][name] = { }
rooms[room][name]['drow'] = drow
rooms[room][name]['yawn'] = yawn
rooms[room][name]['pos'] = pos
rooms[room][name]['number'] = number
rooms[room][name]['docopen'] = docopen
# When particular student joined the room
rooms[room][name]['SessionStart'] = time.time()
rooms[room][name]['avgdrow'] = rooms[room][name]['drow']
# Current Average
rooms[room][name]['avgyawn'] = rooms[room][name]['yawn']
rooms[room][name]['avgpos'] = rooms[room][name]['pos']
rooms[room][name]['avgdocopen'] = rooms[room][name]['docopen']
rooms[room][name]['lastavgdrow'] = 0
rooms[room][name]['lastavgyawn'] = 0
rooms[room][name]['lastavgpos'] = 0 # Last 5 second average
rooms[room][name]['lastavgdocopen'] = 0
rooms[room][name]['update'] = 1 # Tells js to update values
rooms[room][name]['last5'] = time.time()
rooms[room][name]['count'] = 1
rooms[room][name]['drow_val'] = drow_val
rooms[room][name]['paused'] = 0
rooms[room][name]['pavgdrow'] = rooms[room][name]['avgdrow']
rooms[room][name]['pavgyawn'] = rooms[room][name]['avgyawn']
rooms[room][name]['pavgpos'] = rooms[room][name]['avgpos']
# Will be used to calculate last5 second average
rooms[room][name]['pavgdocopen'] = rooms[room][name]['avgdocopen']
rooms[room][name]['pcount'] = rooms[room][name]['count']

```

else:

```

rooms[room] = { }
rooms[room][name] = { }
rooms[room]['class&'] = { }

```

```

# For Average of Class
rooms[room]['class&']['Cdrow'] = 0
rooms[room]['class&']['Cyawn'] = 0
rooms[room]['class&']['Cpos'] = 0 # Initially everything is zero
rooms[room]['class&']['Cdocopen'] = 0

# time in seconds when room was made
rooms[room]['class&']['ClassStartTime'] = time.time()
rooms[room]['class&']['ClassEndTime'] = 0

# For Room Mader ->Teacher
rooms[room][name]['drow'] = drow
rooms[room][name]['yawn'] = yawn
rooms[room][name]['pos'] = pos
rooms[room][name]['number'] = number
rooms[room][name]['docopen'] = docopen
rooms[room][name]['avgdrow'] = rooms[room][name]['drow']

# Current Average
rooms[room][name]['avgyawn'] = rooms[room][name]['yawn']
rooms[room][name]['avgpos'] = rooms[room][name]['pos']
rooms[room][name]['avgdocopen'] = rooms[room][name]['docopen']
rooms[room][name]['lastavgdrow'] = 0
rooms[room][name]['lastavgyawn'] = 0
rooms[room][name]['lastavgpos'] = 0 # Last 5 second average
rooms[room][name]['lastavgdocopen'] = 0
rooms[room][name]['drow_val'] = drow_val
rooms[room][name]['paused'] = 0
rooms[room][name]['update'] = 1 # Tells js to update values
rooms[room][name]['last5'] = time.time()
rooms[room][name]['count'] = 1
rooms[room][name]['pavgdrow'] = rooms[room][name]['avgdrow']
rooms[room][name]['pavgyawn'] = rooms[room][name]['avgyawn']
rooms[room][name]['pavgpos'] = rooms[room][name]['avgpos']

# Will be used to calculate last5 second average

```

```

rooms[room][name]['pavgdocopen'] = rooms[room][name]['avgdocopen']

rooms[room][name]['pcount'] = rooms[room][name]['count']

d = {"Dictionary": rooms}

return jsonify(d)

if __name__ == '__main__':
    app.run(debug=True, host='0.0.0.0')

```

2. Face Detection:

The postFile function sends the image blob as form data using XHR. The function also takes an optional threshold value, which can be included in a data tag. Once the form is set, XHR is used to send it and wait for a response. The response is used to draw rectangles around detected objects in the canvas. The clearRect function is used to clear the canvas before drawing new rectangles. The face_encodings function takes an image as input and generates the face embedding for each face region by looping over the facial landmarks. The function returns a list containing the face embeddings for each face region in the image. The coordinates in the objects object are in percentage units of the image size, so they need to be converted to pixel dimensions before drawing rectangles in the canvas. If the mirror parameter is enabled, the x-axis is flipped to match the flipped mirror view of the video stream.

To build the face recognition system, we generate face embeddings for each image in the dataset and store them in a dictionary. The keys of the dictionary are the names of the people in the dataset and the values are lists of face embeddings for each image of the person. Finally, we save the dictionary to disk.

Now that we have our helper functions, we can start building our face recognition system.

We will start by generating the face embeddings for each image in our dataset. We will then store the face embeddings in a dictionary. The keys of the dictionary will be the names of each person in our dataset and the values will be a list of face embeddings for each image of the person.

Finally, we will save the dictionary to disk.

Sample Codes:

```

from scipy.spatial import distance

from imutils import face_utils, resize

from dlib import get_frontal_face_detector, shape_predictor

```

```

import cv2

import numpy as np

class FaceAction:

    tot = 0

    detect = get_frontal_face_detector()

    predict = shape_predictor("shape_predictor_68_face_landmarks.dat")

    (lStart, lEnd) = face_utils.FACIAL_LANDMARKS_68_IDXS["left_eye"]

    (rStart, rEnd) = face_utils.FACIAL_LANDMARKS_68_IDXS["right_eye"]

    (mStart, mEnd) = face_utils.FACIAL_LANDMARKS_68_IDXS["mouth"]

    K = [6.5308391993466671e+002, 0.0, 3.1950000000000000e+002,

        0.0, 6.5308391993466671e+002, 2.3950000000000000e+002,

        0.0, 0.0, 1.0]

    D = [7.0834633684407095e-002, 6.9140193737175351e-002,

        0.0, 0.0, -1.3073460323689292e+000]

    cam_matrix = np.array(K).reshape(3, 3).astype(np.float32)

    dist_coeffs = np.array(D).reshape(5, 1).astype(np.float32)

    object_pts = np.float32([[6.825897, 6.760612, 4.402142],

        [1.330353, 7.122144, 6.903745],

        [-1.330353, 7.122144, 6.903745],

        [-6.825897, 6.760612, 4.402142],

        [5.311432, 5.485328, 3.987654],

        [1.789930, 5.393625, 4.413414],

        [-1.789930, 5.393625, 4.413414],

        [-5.311432, 5.485328, 3.987654],

        [-2.774015, -2.080775, 5.048531],

        [0.000000, -3.116408, 6.097667],

        [0.000000, -7.415691, 4.070434]])

    reprojectsrc = np.float32([[10.0, 10.0, 10.0],

        [10.0, 10.0, -10.0],

```



```
[10.0, -10.0, -10.0],  
[10.0, -10.0, 10.0],  
[-10.0, 10.0, 10.0],  
[-10.0, 10.0, -10.0],  
[-10.0, -10.0, -10.0],  
[-10.0, -10.0, 10.0]])
```

```
def eye_aspect_ratio(self, eye):  
    A = distance.euclidean(eye[1], eye[5])  
    B = distance.euclidean(eye[2], eye[4])  
    C = distance.euclidean(eye[0], eye[3])  
    ear = (A + B) / (2.0 * C)  
    return ear
```

```
def mouth_aspect_ratio(self, mouth):  
    A = distance.euclidean(mouth[13], mouth[19])  
    B = distance.euclidean(mouth[14], mouth[18])  
    C = distance.euclidean(mouth[15], mouth[17])  
    D = distance.euclidean(mouth[12], mouth[16])  
    mar = (A + B + C) / (2.0 * D)  
    return mar
```

```
def drowsy(self, frame):  
    frame = resize(frame, width=450)  
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)  
    subjects = self.detect(gray, 0)  
    self.tot = len(subjects)  
    # print(len(subjects))  
    # print(self.tot)  
    if (len(subjects) == 0):  
        return 1  
    for subject in subjects:
```

```

shape = self.predict(gray, subject)
shape = face_utils.shape_to_np(shape)
leftEye = shape[self.lStart:self.lEnd]
rightEye = shape[self.rStart:self.rEnd]
leftEAR = self.eye_aspect_ratio(leftEye)
rightEAR = self.eye_aspect_ratio(rightEye)
ear = (leftEAR + rightEAR) / 2.0
return ear

```

```

def yawn(self, frame):
    frame = resize(frame, width=450)
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    subjects = self.detect(gray, 0)
    if (len(subjects) == 0):
        return 0
    for subject in subjects:
        shape = self.predict(gray, subject)
        shape = face_utils.shape_to_np(shape)
        mouth = shape[self.mStart:self.mEnd]
        mar = self.mouth_aspect_ratio(mouth)
    return mar

```

```

def get_head_pose(self, shape, object_pts, cam_matrix, dist_coeffs, reprojectsrc):
    image_pts = np.float32([shape[17], shape[21], shape[22], shape[26], shape[36],
                             shape[39], shape[42], shape[45], shape[31], shape[35],
                             shape[48], shape[54], shape[57], shape[8]])

    _, rotation_vec, translation_vec = cv2.solvePnP(
        object_pts, image_pts, cam_matrix, dist_coeffs)

    reprojectdst, _ = cv2.projectPoints(reprojectsrc, rotation_vec, translation_vec,
cam_matrix,

```

```

        dist_coeffs)

    reprojectdst = tuple(map(tuple, reprojectdst.reshape(8, 2)))

    rotation_mat, _ = cv2.Rodrigues(rotation_vec)
    pose_mat = cv2.hconcat((rotation_mat, translation_vec))
    _, _, _, _, _, _, euler_angle = cv2.decomposeProjectionMatrix(pose_mat)

    return reprojectdst, euler_angle

def head_pose(self, frame):

    face_rects = self.detect(frame, 0)

    if(len(face_rects) > 0):
        shape = self.predict(frame, face_rects[0])
        shape = face_utils.shape_to_np(shape)

        _, euler_angle = self.get_head_pose(
            shape, self.object_pts, self.cam_matrix, self.dist_coeffs, self.reprojectsrc)

        if(-10 <= euler_angle[2, 0] and euler_angle[2, 0] <= 10):
            return 0

        else:
            return 1

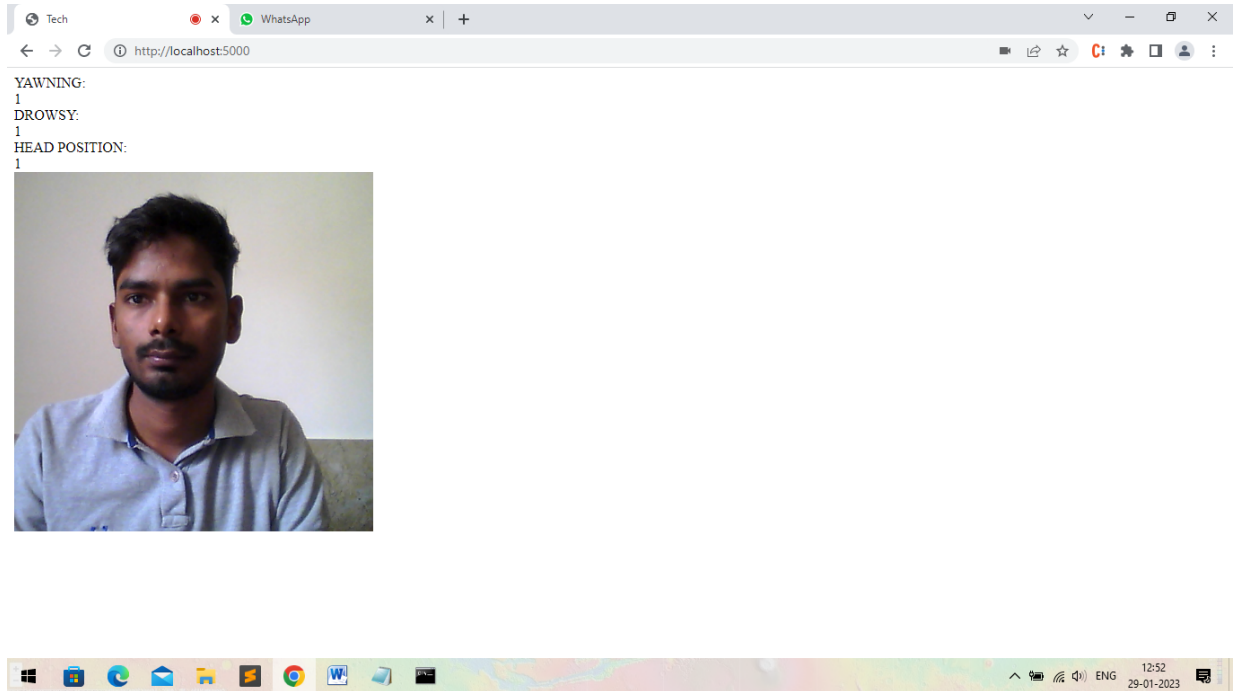
    else:
        return 1

def run_frame(self, frame):

    return (self.drowsy(frame), self.yawn(frame), self.head_pose(frame), self.tot)

```

Screen shot:



API SERVER PAGE

3. Client Server :

The setup described above runs every frame through the server, which can consume a lot of CPU power, even when there is no activity in the video stream. To improve efficiency, we can limit the API calls and only invoke them when there is new activity in the video stream. To accomplish this, we can modify `objDetect.js` and create a new file called `objDetectOnMotion.js`.

The `objDetectOnMotion.js` file is mostly the same as `objDetect.js`, but it includes two new functions. The first function, called `sendImageFromCanvas()`, only sends an image if it has changed within a given framerate, which is determined by the `updateInterval` parameter. We will use a new canvas and context for this.

Further research has revealed that other algorithms typically convert images to grayscale because color is not a good indicator of motion. Additionally, applying a Gaussian blur can smooth out encoding variances. Fippo suggested exploring Structural Similarity algorithms as used by the `test.webrtc.org` to detect video activity.

Sample Codes:

```
from flask import Flask, render_template, request, url_for, redirect, jsonify
import json
app = Flask(__name__)
```

```
userdata = 0
```

```
@app.route('/')
def hello_world():
    return render_template('index.html')
```

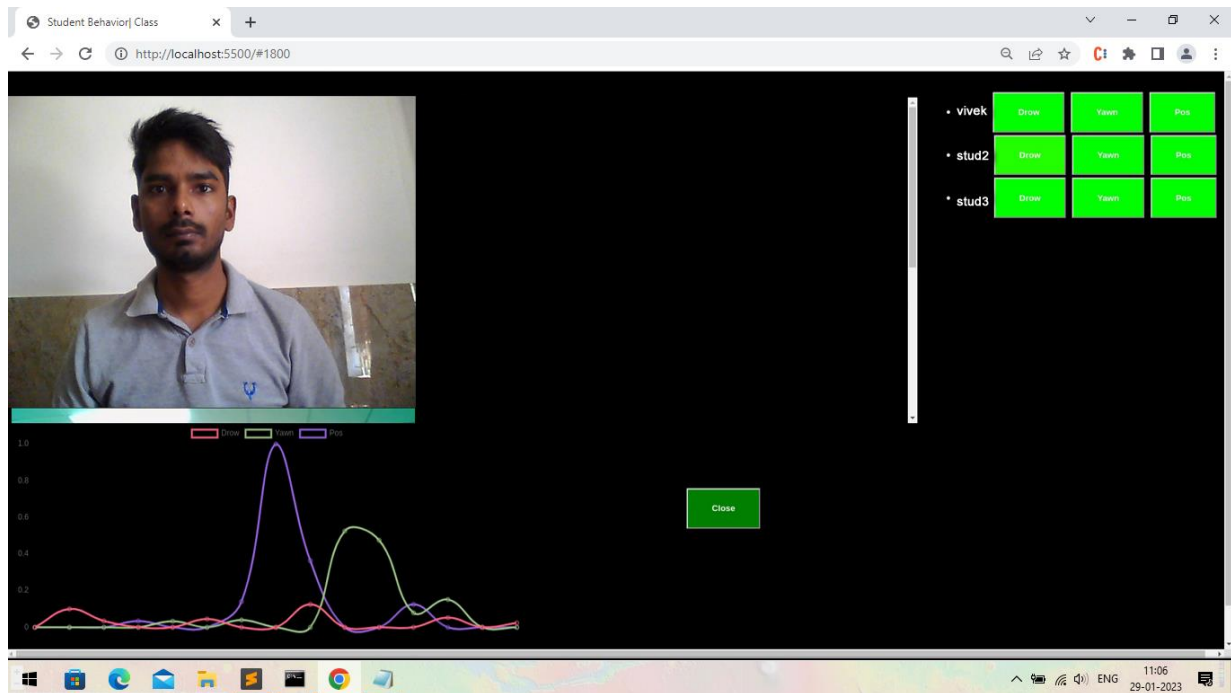
```
@app.route('/new')
def new():
    print("Here we are")
    return render_template('result.html')
    # return jsonify(userdata)
```

```
@app.route('/getdata', methods=['GET'])
def getdata():
    print("here getedata")
    return jsonify(userdata)
```

```
@app.route('/result', methods=['GET', 'POST'])
def your_func():
    print(request.form)
    # print(type(request.form['data']))
    # print(json.loads(request.form))
    global userdata
    userdata = request.form
    # print(request.method)
    # print(request.form)
    # if (request.method == 'POST'):
    #     print("here I am")
    # return render_template('result.html')
    return redirect(url_for('new'))
```

```
if __name__ == '__main__':
    app.run(debug=True, host='0.0.0.0', port=5500)
```

Screenshot:



STUDENT DASHBOARD

3. Visualizing Student Behavior and Performance

we can show the general attentiveness of all the students in the class to the teacher, highlighting the durations of the lecture where the students felt the most boredom or where most students had doubts, giving the lecturer an analysis on his/her performance and helping him in further improving the course. On the Students' side, he/she would receive the feedback about his/her own concentration for the whole duration. Besides these features, the students and teachers, both would have a dashboard - where their past performance would be stored and also, they can add a to-do list for them to finish for a particular day.

Sample Code:

```
<!DOCTYPE html>
```

```
<html id="home" lang="en">
```

```
<head>
```

```
<link rel="stylesheet" href="static/css/res.css">
```

```
<link rel="stylesheet" href="static/bootstrap.min.css">
```

```
<link rel="stylesheet" type="text/css"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/css/bootstrap.min.css">
```

```
<script src="static/jquery.slim.min.js"></script>
```

<script src="static/bootstrap.min.js"></script>

<script src="static/popper.min.js"></script>

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">

<title>Your Result</title>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

</head>

<body style="background-color: lavender;">

<div class="topnav" id="myTopnav">

Home

News

Contact

About

<i class="fa fa-bars"></i>

</div>

<div class="row">

<div class="col-lg-6 col-sm-4">

<canvas id="myPieChart1"></canvas>

</div>

<div class="col-lg-6 col-sm-4"><canvas id="myPieChart2"></canvas></div>

<div class="w-100 d-none d-md-block"></div>

<div class="col-lg-6 col-sm-4"><canvas id="myPieChart3"></canvas></div>

<div class="col-lg-6 col-sm-4"><canvas id="myPieChart4"></canvas></div>

</div>

<div>

```

<h3>Various Other Data</h3>

<h5 id="mydata"></h5>

</div>

<script src="static/chart.js"></script>

<script src="static/res.js"></script>

<script src="https://code.jquery.com/jquery-2.1.4.js"></script>

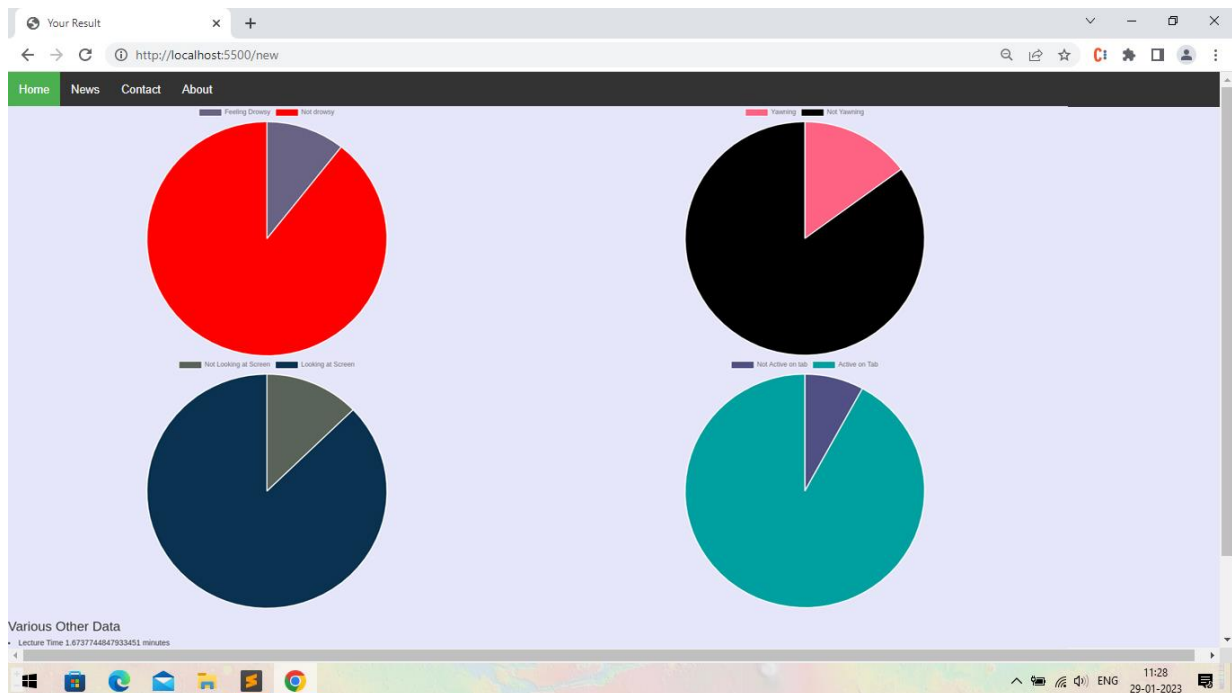
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/js/bootstrap.min.js"></script>

</body>

</html>

```

Screenshot:



Visualizing Student Behavior and Performance

6.6 RESULT ANALYSIS & EVALUATION METRICS

6.6.1 Sample Test Cases:

Here are some test cases that can be used to verify the functionality of the E-education project:

- **One-to-Many Communication:** Verify that teachers can deliver live classes to multiple students in the local network without any lag or delay in video communication.
- **Facial Detection:** Verify that the system can accurately detect the faces of teachers and students in the video stream, even in low light conditions or with varying camera angles.
- **Real-Time Feedback:** Verify that the system can generate real-time feedback using charts that accurately reflect the level of understanding and engagement of students during the lecture.
- **Alert Systems:** Verify that the system can generate alerts to teachers when students are not engaging or participating in the lecture, such as when they are browsing other websites or have minimized the video window.
- **Summary of Video Lecture:** Verify that the system can generate accurate summaries of the video lecture that can be accessed by both the teacher and the students for review.
- **ML Model Analysis:** Verify that the ML model can accurately analyze the attentiveness of students during the lecture and identify periods where students were most bored or had the most doubts.
- **Course Improvement:** Verify that the system can provide teachers with actionable insights to improve their performance and the course material, such as highlighting areas where students struggled or where the lecture was not engaging enough.
- **Compatibility:** Verify that the E-education project is compatible with different browsers, operating systems, and devices, to ensure that all students can access the course material and live classes without any issues.

6.6.2 Performance Metrics:

We conducted a re-evaluation of our target detection results using images with various backgrounds. We noticed that there were cases where the face detection failed due to changes in the background, and identified image brightness as a key factor in improving the accuracy of our algorithm. We were able to achieve real-time image processing speed using a selected computer configuration while performing real-time webcam tests. However, we have not fully evaluated the effectiveness of our proposed method due to the limited number of gestures. We can assess the accuracy of our face landmark detection and recognition steps.

Performance metrics that can be used to evaluate the target detection algorithm:

- **Background Variations:** The algorithm's ability to detect targets under different backgrounds can be measured by using images with various backgrounds and evaluating the algorithm's performance under different lighting and environmental conditions.
- **Image Brightness:** The impact of image brightness on target detection accuracy can be measured by analyzing the algorithm's performance under varying levels of brightness. This can be done by modifying the brightness of the test images and evaluating the algorithm's accuracy under different lighting conditions.
- **Real-time Performance:** The real-time performance of the algorithm can be measured by evaluating the processing speed of the algorithm while performing real-time webcam tests. This can be measured in terms of frames per second (FPS) and latency.
- **Hardware Configuration:** The hardware configuration used to run the algorithm can also be evaluated as a performance metric. This can include the CPU and GPU specifications, memory usage, and other system resources required to achieve real-time processing speed.
- **User Experience:** The overall user experience of the algorithm can also be evaluated as a performance metric. This can include factors such as ease of use, user interface design, and user satisfaction levels, which can impact the adoption and effectiveness of the algorithm in real-world scenarios.

CONCLUSIONS AND FUTURE WORK

7.1 Conclusion:

The aim of this project is to enhance the quality of E-education in a local network by providing one-to-many video communication, live classes by teachers to students in a college or university network. To make the teaching more analysis-oriented, the project incorporates technologies such as Facial Detection using Deep Learning, real-time Feedback using Charts, and Alert systems. A summary of the video lecture is created for both the teacher and the students. At the end of the lecture, the ML Model's data is used to show the general attentiveness of all the students in the class to the teacher, highlighting the periods of the lecture where the students were most bored or had the most doubts, providing the lecturer with an analysis of his/her performance and assisting in further course improvement.

7.2 Future Work:

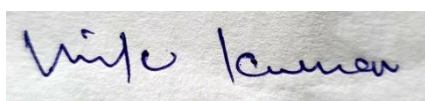
Some future work ideas that can be pursued:

- **Integration of Virtual Reality:** The project can be extended by integrating virtual reality technologies to create a more immersive learning experience. This would allow students to visualize concepts in three dimensions and interact with digital objects in a more intuitive manner.
- **Natural Language Processing:** Natural Language Processing (NLP) can be used to enhance the quality of feedback provided to students. This can be achieved by analyzing the questions asked by students during the lecture, identifying the areas where students are struggling, and providing targeted feedback to students.
- **Integration with Learning Management Systems:** The project can be integrated with Learning Management Systems (LMS) to provide a seamless experience for students and teachers. This would allow teachers to easily create and manage online courses, while students can access course materials and participate in live classes from a single platform.
- **Multimodal Learning:** The project can be extended to support multimodal learning, which involves the use of multiple modalities such as audio, video, and text to convey information. This would enable students with different learning styles to engage with the material in a way that suits them best.
- **Personalized Learning:** The ML model can be extended to provide personalized learning recommendations to students based on their learning history and performance. This would allow students to receive tailored recommendations on which topics to focus on, as well as personalized feedback on their performance.
- **Collaboration Tools:** The project can be extended to include collaboration tools that enable students to work together on group assignments and projects. This would allow students to develop teamwork skills and learn from each other in a more collaborative environment.

References:

- [1] D. Yang et al. "An Emotion Recognition Model Based on Facial Recognition in Virtual Learning Environment" *Procedia Computer Science* 6th International Conference on Smart Computing and Communications, ICSCC 2017, 7-8 December 2017, Kurukshetra, India 125 (2018) 2–10
- [2] X. You, J. Xu and W. Yuan et al "Multi-view common component discriminant analysis for cross-view classification" in *Proc. Elsevier*- 92 (2019) 37–51
- [3] ahmane and J. Meunier, "Emotion recognition using dynamic gridbased HoG features," 2011 IEEE International Conference on Automatic Face & Gesture Recognition (FG), 2011, pp. 884-888, doi: 10.1109/FG.2011.5771368.
- [4] D. Yang, Abeer Alsadoon, P.W.C. Prasad, A.K. Singh, A. Elchouemi, An Emotion Recognition Model Based on Facial Recognition in Virtual Learning Environment, *Procedia Computer Science*, Volume 125, 2018.
- [5] Deepak Kumar Jain, Pourya Shamsolmoali, Paramjit Sehdev, Extended deep neural network for facial emotion recognition, *Pattern Recognition Letters*, Volume 120, 2019, Pages 69-74, ISSN 0167-8655
- [6] Jiankang Deng, Jia Guo , Jing Yang , Niannan Xue "ArcFace: Additive Angular Margin Loss for Deep Face Recognition" *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, VOL. 44, NO. 10, OCTOBER 2022
- [7] Egils Avots¹ · Tomasz Sapiński³ · Maie Bachmann "Audiovisual emotion recognition in wild" *Machine Vision and Applications* (2019) 30:975–985 <https://doi.org/10.1007/s00138-018-0960-9>
- [8] Hamed Monkaresi, Nigel Bosch, Rafael A. "Automated Detection of Engagement Using Video-Based Estimation of Facial Expressions and Heart Rate" *IEEE TRANSACTIONS ON AFFECTIVE COMPUTING*, VOL. 8, NO. 1, JANUARY-MARCH 2017
- [9] Woo-Han Yun , Dongjin Lee "Automatic Recognition of Children Engagement from Facial Video Using Convolutional Neural Networks" *IEEE TRANSACTIONS ON AFFECTIVE COMPUTING*, VOL. 11, NO. 4, 1949-3045-OCTOBER-DECEMBER 2020.
- [10] P.Esther Rani, S.Velmurugan, "Behavioral Analysis of Students by Integrated Radial Curvature and Facial Action Coding System using DCNN" 978-1-6654-0816-5/22/\$31.00 ©2022 IEEE
- [11] Minsu Jang ETRI, Cheonshu Park ETRI "Building an Automated Engagement Recognizer based on Video Analysis" author/owner(s). HRI'14, March 3–6, 2014, Bielefeld, Germany. ACM 978-1-4503-2658-2/14/03. <http://dx.doi.org/10.1145/2559636.2563687>.
- [12] Sahla K. S. and T. Senthil Kumar "Classroom Teaching Assessment Based on Student Emotions" *Intelligent Systems Technologies and Applications 2016, Advances in Intelligent Systems and Computing* 530, DOI 10.1007/978-3-319-47952-1_37
- [13] Hariharan S, J Daniel Pushparaj, Muthukumaran Malarvel "Computer Vision based Student Behavioral Tracking and Analysis using Deep Learning" Department of Computer Science and Engineering, Hindustan Institute of Technology and Science, Chennai, Tamil Nadu- 978-1-6654-7971-4/22/\$31.00 ©2022 IEEE
- [14] Khawlah Altuwairqi¹ · Salma Kammoun Jarraya "Student behavior analysis to measure engagement levels in online learning environments" *Signal, Image and Video Processing* (2021) 15:1387–1395 <https://doi.org/10.1007/s11760-021-01869-7>
- [15] Kyu-Seob Song¹ Young-Hoon Nho² "Decision-Level Fusion Method for Emotion Recognition using Multimodal Emotion Recognition Information" 2018 15th International Conference on Ubiquitous Robots (UR) Hawaii Convention Center, Hawai'i, USA, June 27-30, 2018

- [16] Archana Sharma, Vibhakar Mansotra “Deep Learning based Student Emotion Recognition from Facial Expressions in Classrooms” International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249-8958 (Online), Volume-8 Issue-6, August, 2019
- [17] Mohamed Dahmane and Jean Meunier “Emotion Recognition using Dynamic Grid-based HoG Features” J. Meunier Department of Computer Science and Operations Research (DIRO), University of Montreal, 2920 Chemin de la tour, Montreal, Quebec, Canada, H3C 3J7 meunier@iro.umontreal.ca
- [18] Pierluigi Carcagnì*,†, Marco Del Coco† , Marco Leo “Facial expression recognition and histograms of oriented gradients: a comprehensive study” Institute of Applied Sciences and Intelligent Systems, Carcagnì et al. SpringerPlus (2015) 4:645 DOI 10.1186/s40064-015-1427-3
- [19] Amit Chavda, Amit Chavda “Multi-Stage CNN Architecture for Face Mask Detection” 2021 6th International Conference for Convergence in Technology (I2CT) Pune, India. Apr 02-04, 2021
- [20] ZUOLIN DONG 1 , JIAHONG WEI 2 , XIAOYU CHEN 1 , AND PENGFEI ZHENG 3” Face Detection in Security Monitoring Based on Artificial Intelligence Video Retrieval Technology, 10.1109/ACCESS.2020.2982779
- [21] Soroosh Parsai dept. of electrical and computer engineerin University of Windsor Windsor, Canada, d Majid Ahmadi “A Low Error Face Recognition System Based on A New Arrangement of Convolutional Neural Network and Data Augmentation” TENCON 2022 - 2022 IEEE Region 10) | 978-1-6654-5095-9/22/\$31.00 ©2022 IEEE | DOI: 10.1109/TENCON55691.2022.9978010
- [22] YIBO CAO , SHUN LIU, PENGFEI ZHAO, AND HAIWEN ZHU School of Software, South China Normal University, Guangdong 510631, China “RP-Net: A PointNet++ 3D Face Recognition Algorithm Integrating RoPS Local Descriptor” DOI 10.1109/ACCESS.2022.3202216
- [23] DANISH ALI , IMRAN TOUQIR , ADIL MASOOD SIDDIQUI, JABEEN MALIK , AND MUHAMMAD IMRAN Department of Electrical Engineering,” Face Recognition System Based on Four State Hidden Markov Model” National University of Sciences and Technology, Islamabad 46000, Pakistan DOI r 10.1109/ACCESS.2022.3188717
- [24] AHMED RIMAZ FAIZABADI 1,2, (Member, IEEE), HASAN FIRDAUS BIN MOHD ZAKI “Efficient Region of Interest Based Metric Learning for Effective Open World Deep Face Recognition Applications” Department of Mechatronics, Kulliyah of Engineering, International Islamic University Malaysia, Gombak, Kuala Lumpur 53100, Malaysia Digital Object Identifier 10.1109/ACCESS.2022.3192520
- [25] Cedric Nimpa Fondje , Graduate Student Member, IEEE, Shuowen Hu, Member, IEEE, and Benjamin S. Riggan “Learning Domain and Pose Invariance for Thermal-to-Visible Face Recognition” unl.edu). Shuowen Hu is with the U.S. Army Combat Capabilities Development Command, Army Research Laboratory, Adelphi, MD 20783 USA, Digital Object Identifier 10.1109/TBIOM.2022.3223055



Digital Signature (Student)

Digital Signature (Guide)

X

Dr.Vijayan.R,Associate Proessor
SITE,VIt,Vellore