**A Project Report on**

# ATTENTION SPAN DETECTION IN ONLINE INSTRUCTOR LED SESSIONS USING BIGDATA

**Submitted to**

**Jawaharlal Nehru Technological University, Hyderabad**

**In partial fulfillment of requirements for the award of the degree of**

**BACHELOR OF TECHNOLOGY**

**In**

**COMPUTER SCIENCE AND ENGINEERING**

**By**


**POOJA AILANI (16BD1A05A6)**

**GAYATHRI SARAPU (16BD1A05B3)**


**Under the esteemed guidance of**

**A.SRINIVAS RAO**

Assistant Professor

Department of CSE



**Department of Computer Science and Engineering KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY Approved by AICTE, Affiliated to JNTUH 3-5-1206, Narayanaguda, Hyderabad - 500029 2019 – 2020**

# KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

Approved by AICTE, Affiliated to JNTU, Hyderabad

3-5-1206, Narayanaguda, Hyderabad - 500029.

## CERTIFICATE

This is to certify that the project entitled "**ATTENTION SPAN DETECTION IN ONLINE INSTRUCTOR LED SESSIONS USING BIGDATA"** being submitted by **POOJA AILANI(16BD1A05A6),GAYATHRI SARAPU(16BD1A05B3),** student of **Keshav Memorial Institute of Technology, JNTUH** in partial fulfillment of requirements of the award of the Degree of **Bachelor of Technology in Computer Science and Engineering** as a specialization is a record of bonafide work carried out by them under my guidance and supervision in the academic year 2019 – 2020.

**GUIDE**
**Mr. A SRINIVAS RAO**
Assistant Professor

**HOD**
**Dr. S. Padmaja**
**CSE DEPARTMENT**

**Submitted for the Project Viva Voce examination held on …….**

**EXTERNAL EXAMINER**

# DECLARATION

We hereby declare that the project report entitled **"ATTENTION SPAN DETECTION IN ONLINE INSTRUCTOR LED SESSIONS USING BIGDATA"** is done in the partial fulfillment for the award of the Degree of Bachelor of Technology in Computer Science and Engineering affiliated to Jawaharlal Nehru Technological University, Hyderabad. This project has not been submitted anywhere else.

**POOJA AILANI (16BD1A05A6)**

**GAYATHRI SARAPU (16BD1A05B3)**

# ACKNOWLEDGEMENT

We take this opportunity to thank all the people who have rendered their full support to our project work.

We render our thanks to **Dr. Maheshwar Dutta**, M.Tech., Ph.D.,Principal who encouraged us to do the Project.

We are grateful to **Mr. Neil Gogte**, Director for facilitating all the amenities required for carrying out this project.

We express our sincere gratitude to **Mr. S. Nitin**, Director and **Mrs. Deepa Ganu**, Dean Academic for providing an excellent environment in the college.

We are also thankful to **Dr. S. Padmaja**, Head, Department of CSE for providing us with both time and amenities to make this project a success within the given schedule.

We are also thankful to our guide **A.SRINIVAS RAO**, for his/her valuable guidance and encouragement given to us throughout the project work.

We would further like to thank the entire CSE Department faculty, who helped us directly and indirectly in the completion of the project.

We sincerely thank our friends and family for their constant motivation during the project work.

**POOJA AILANI (16BD1A05A6)**

**GAYATHRI SARAPU (16BD1A05B3)**

# CONTENTS

# ABSTARCT

Online education is a flexible instructional delivery system that gives educators an opportunity to reach students who may not be able to enrol in a traditional classroom course and supports students who need to work on their own schedule and at their own pace. This concept falls under smart education domain. With the exponential technological advances, anything could be instrumented, interconnected, and infused with intelligent design, so is education .Smart education is a concept that describes learning in digital age and because of its growing popularity, it is defined as self-directed, motivated, adaptive, resource-enriched, and technology-embedded.

An online lecture is an educational lecture designed to be posted online. Learners may access online lectures posted on their designated websites anywhere in the world, at any time they wish, as long as they have an internet connection. Learners are expected to go through videos and attend the instructor led sessions on a weekly basis. The agenda of instruction led sessions is to do a quick debrief of video lectures for the week.

Even though online education has excelled throughout, it still has its own drawbacks. One of the major drawback is student's lack of attention. Our problem statement is to create a mechanism to monitor and measure the attention span of the learners in the instructor led online sessions and dashboard that quantifies the participation for each participant. In order to solve this problem machine learning can be used along with big data to analyse the huge dataset on apache spark framework for our project.

Source of Dataset: https://iith.ac.in/~daisee-dataset/dataset/

# LIST OF FIGURES

# LIST OF SCREENSHOTS

# CHAPTER-1

# 1. INTRODUCTION

## 1.1 HISTORY

Online education had its beginnings in the late nineteen hundred; the concept of distance learning first came into practice in the mid nineteenth century when the U.S. Postal Service was developed. The notion of reliable, long-distance correspondence led to the development and implementation of correspondence colleges, where instructional missives would be distributed through the postal service between students and professors. Today, long-distance education programs have become more sophisticated and accessible due to the advancement of the web and digital technology. Elite institutions around the world now offer open courseware, online degrees, and online classes that are both legitimizing and popularizing the idea of education from a computer.

Online Education ties together several historical threads like computers, distance learning, and telecommunications. The concept is over one hundred and seventy years old and it origins in a corresponding course. Online education overcomes the biggest drawback of correspondence course, mail transit time and allows the students to interact not only with the instructor but with the other students in the class in real time. Online education has emerged triumphant and has been proven to be viable education choice in its every right. Considering all the factors related to online education the most important drawback of eLearning is student's attention span which has proven to be a major hindrance in terms of student's academic performance. In order to deal with this problem, and to make online education more effective and reliable we use learning analytics.

Learning Analytics is the measurement, collection, analysis and reporting of data about learners and their contexts, for purposes of understanding and optimizing learning and the environments. Learning analytics is both an academic field and commercial marketplace which have taken rapid shape over the last decade. People have been researching learning and teaching, tracking student progress, analyzing school or university data, designing assessments and using evidence to improve teaching and learning for a long time. Learning Analytics builds on these well established disciplines, but seeks to exploit the new opportunities once we capture new forms of digital data from students' learning activity, and use computational analysis techniques from data science and AI.

Learning Analytics provides researchers with exciting new tools to study teaching and learning. Moreover, as data infrastructures improve from data capture and analysis, to visualization and recommendation we can close the feedback loop to learners, offering more timely, precise, actionable feedback. In addition, educators, instructional designers and institutional leaders gain new insights once the learning process is persistent and visible.

## 1.2 DOMAIN

With the increasing technological advances, anything could be instrumented, interconnected, and infused with intelligent design and so is education. Our domain for this project is smart education. The development in technology has encouraged the learners to learn more effectively, efficiently and comfortably. Learners can access any sort of digital resources through smart devices when it is connected to a wireless network. Smart education is a concept that describes learning in digital age, has gained increased attention. There is no clear and unified definition of smart learning so far. Multidisciplinary researchers and educational professionals are continuously discussing the concept of smart learning.

MEST presented the features of smart learning that is defined as self-directed, motivated, adaptive, resource-enriched, and technology-embedded. Lee-et-al proposed that the features of smart learning include formal and informal learning, social and collaborative learning, personalized and situated learning, and application and content focus. Smart learning environment not only enables learners to access ubiquitous resources and interact with learning systems anytime and anywhere, but also provides the necessary learning guidance, suggestions or supportive tools to them in the right form, at the right time and in the right place. Koper proposed that smart learning environments are defined as physical environments that are enriched with digital, context-aware and adaptive devices, to promote better and faster learning.

Usually, ‘smart’ in smart education refers to intelligent, personalized and adaptive. But for different entities and/or educational situations, the meaning of ‘smart’ has different definitions. For learner, ‘smart’ refers to wisdom and intelligence. For educational technology, ‘smart’ refers to accomplish its purpose effectively and efficiently. For educational environment, ‘smart’ refers to engaging, intelligent and scalable. In order to foster the learners’ performance, there are four instructional strategies.

These strategies include class-based differentiated instruction, group-based collaborative learning, individual- based personalized learning and mass-based generative learning. All these strategies encompass formal and informal learning, in both the real and the digital world.

Based on the research framework of smart education, there are two case studies. One is flipped classroom project that integrated smart pedagogies and constructed smart learning environments for students. Another pilot project is called “Online J classroom” that also integrated smart pedagogies into learning process to realize precision teaching. To realize personalized learning experience, smart learning environments can provide accurate and rich learning services by using learning analytics.

## 1.3 PURPOSE

Online education has been soaring high these days, it can be possible that the future education will be relying on it .In order to make it more efficient and reliable we must tackle the drawbacks of e-learning. The major issue which is attention span can be handled using learning analytics which can be implemented for a better quality education.

The purpose of our project is to create a mechanism wherein we will be able to predict the attention span of a student in a video lecture.

## 1.4 PROPOSED SYSTEM

The dataset consists of videos of ninety five students, the video on the whole is about 30 hours and each video is divided into exactly ten seconds. Our project starts by converting these videos to data frames; from each video we extract ten images as data frames. These data frames are given labels and converted into arrays for further processing.

The model is created using ResNet-50 which is a convolutional neural network and is saved for further testing. For displaying the report we use sklearn package and import various models like classification_report and labelBinarizer to store the labels.

The testing process requires loading the model and labels that were saved. A video is given as an input to detect the attention span of the student. The input video is converted to data frames and each data frame is processed. The classified output label will be displayed along with the video. Since the dataset is very large there are lot of elements which needs to be taken care of. It is not possible to execute in a normal laptop or a computer solely. There is a high possibility that the system may get overloaded resulting in a crash. In order to solve this problem we are using Hadoop cluster.

The Hadoop Distributed File System acts as storage medium and can be accessed more easily. It is also more reliable and faster compared to our laptop or personal computer.

## 1.5 SCOPE

When it comes to Online Education, one must consider the fact that online education is not limited to only one country and institution. This is considered an education without boundaries, which allows the students to study their chosen subjects at their ease. In order to increase the efficiency and tackle various drawbacks we need to implement learning analytics, this will help in improving student's ability to learn as well as the infrastructure of online education. In this digital age it is necessary that we incorporate various methods of machine learning to achieve maximum students' attention and ensure better results.

# CHAPTER-2

# 2. SOFTWARE REQUIREMENT SPECIFICATIONS

Software Requirement Specification (SRS) is the starting point of the software developing activity. As the systems grew more complex it became evident that the goal of the entire system cannot be easily comprehended. Hence the need for the requirement phase arose. The software project is initiated by the client needs. SRS is the means of translating the ideas of the minds of clients (the input) into a formal document (the output of the requirement phase). In other words SRS is a formal document that acts as a written agreement between the development team and the customer. This includes gathering, analysing, validating and specifying requirements.

The purpose of the Software Requirement Specification is to reduce the communication gap between the clients and the developers and describe the external behavior of the software or application developed. Software Requirement Specification is the medium though which the client and user needs are accurately specified. It forms the basis of software development. It defines the operations, performance and interfaces and quality assurance requirement of the application or software. The complete software requirements for the system are captured by the SRS.

Specific goals of SRS include:

- Facilitating reviews
- Describing the scope of work
- Providing a reference to software designers
- Providing a framework for testing primary and secondary use cases
- Including features to customer requirements
- Providing a platform for ongoing refinement

The software requirement specification document for attention span detection in instructor led online sessions enlists the functional, non-functional and system requirements.

## 2.1 FUNCTIONAL REQUIREMENTS

A functional requirement defines a system or its component. It defines the functions software must perform. A function can be specified by identifying the state at which data is to be input to the system, its input data domain, the output domain, and the type of processing to be carried on the input data to obtain the output data. Functional requirements define specific behavior or function of the application.

The functional requirements are detailed as follows:

**Converting the video stream into Images**

Using a full HD web camera videos of the students who have enrolled in a course are captured in twenty minutes duration each. All the videos are divided into ten seconds snippets. Each video is then converted into image frames.

**Processing the images**

The images generated from video stream are then individually processed for face detection. Preprocessing includes swapping color channels for OpenCV to Keras compatibility, resizing to 224×224px, random rotations, zooms, shifts, shears, and flips.

**Differentiate between different emotions**

Our dataset consists of labelling of four emotional states engagement, frustration, confusion and boredom. For each of the affective states, there are four labels: (0) very low (1) low (2) high and (3) very high. The system should be able to differentiate between different emotions of the students.

**Predict the emotion of the student when video is given**

When the video of any student is passed for testing the system should be able to predict the emotion state the student possesses.

## 2.2 NON-FUNCTIONAL REQUIREMENTS

A non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviours. These define system attributes such as security, reliability, performance, maintainability, scalability and usability. Especially these are the constraints the system must work within. Failing to meet any one of them can result in systems that fail to satisfy internal business, user or market needs or that do not fulfil mandatory requirements imposed by regulatory or standards agencies.

Some of the non-functional requirements include:

- Should accept huge datasets.
- Should work on any platform.
- Response time for each prediction should be less.
- The software should run smoothly without any glitches.
- Should be adaptable to any level of mobile data/wifi.

## 2.3 SYSTEM REQUIREMENTS

System requirements are the configuration that a system must have in order for a hardware or software application to run smoothly and efficiently. Failure to meet these requirements can result in installation problems or performance problems. The former may prevent a device or application from getting installed, whereas the latter may cause a product to malfunction or perform below expectation or even to hang or crash. System requirements are also known as minimum system requirements.

### 2.3.1 Hardware Requirements

Hardware system requirements often specify the operating system, version, processor type, memory size, available disk space and additional peripherals.

- Processor : intel Core i5
- Hard Disk : 1TB
- RAM : 8GB or more
- GPU : Intel® HD Graphics 520

### 2.3.2 Software requirements

Software system requirements, in addition to the aforementioned requirements, may also specify additional software dependencies (libraries, driver version, framework version). Some software or hardware manufacturers provide an upgrade assistant program that users can download and run to determine whether their system meets a product's requirements.

- Operating System : Windows 8/10, Linux
- Platform : Hadoop
- Technology : Python 3.7
- IDE : Spyder
- UML : ArgoUML

# CHAPTER-3

# 3. LITERATURE SURVEY

**Article 1: Students' Attention Assessment in eLearning based on Machine Learning**

**Author:** Qingshan Deng, Zhilli Wu

**University:** School of Software & Internet of Things Engineering, Jiangxi University of Finance and Economics, Nanchang 330013, Jiangxi, China

**Abstract:** Multimedia-based Electronic learning (eLearning) is an effective method of knowledge transfer. It provides the opportunity that students can use the videos or other materials at any time after they are delivered Multimedia applications provide convenience, but there also exist challenges. One of the challenges is to measure and assess students' attention when they are studying online. This paper presented a framework based on machine learning methods for the measurement of students' attention. The framework employs a Gabor wavelet to extract the eye state features and train the model using support vector machines (SVM) to complete automatic classification on students' eye states. Experiments over thousands of facial photos show that the proposed system reaches a good performance, which has a significant value in real applications.

**Related work:** A framework to measure the student's visual attention automatically was proposed which started working by extracting frames from the video stream of the student's webcam. Each frame is analysed by searching the faces including facial points and eye states. Eyes' states are detected automatically to decide whether they are opened or closed. This detection is based on machine learning algorithms and predictors.

**Figure 3.1 Student's visual attention assessment framework**

Figure 3.1 is the schematic diagram for proposed framework, which include following steps:

1. **Face Detection:** Firstly, they determined whether there is a face or not in an image or video sequence.
2. **Facial point detection:** Using machine learning model 68 facial points were generated.
3. **Eye detection:** After obtaining the facial points, eye area-related points were detected and eye images were extracted.
4. **Eye state classification:** After obtaining the eye photos they are classified as opened or closed.
5. **Attention scoring:** The students' attentiveness based on the face detection and the eye state can be divided into 5 levels: Away, Sleepy, Mind wandering, Satisfactory and Attentive according to the attention scores.

**Article 2: Detecting Mind-Wandering from Eye Movement and Oculomotor Data during Learning Video Lecture**

**Author:** DongMin Jang, IlHo Yang, and SeoungUn Kim

**University:** Korea National University of Education of Elementary Science, Taeseongtabyeon-ro, Gangnae-myeon, Heungdeok-gu, Chungbuk 28173, South Korea

**Abstract:** The purpose of this study was to detect mind-wandering experienced by pre-service teachers while learning video lecture on physics. The lecture was videotaped and consisted of a live lecture in a classroom. They investigated whether oculomotor data and eye movements could be used as a marker to indicate the learner's mind-wandering. Each data was collected in a study in which 24 pre-service teachers (16 females and 8 males) reported self-caught mind-wandering while learning physics video lecture during30minutes. A Tobii Pro Spectrum (sampling rate: 300Hz) was used to capture their eye-gaze during learning Gauss's law course video. After watching video lecture, they interviewed pre-service teachers about their mind-wandering experience. The self-caught method was used to capture the mind-wandering timing of pre-service teachers while learning from video lectures. Accurate mind-wandering segments by comparing fixation duration and saccade count.

Two types of oculomotor data (blink count, pupil size) and nine eye movements (average peak velocity of saccades; maximum peak velocity of saccades; standard deviation of peak velocity of saccades; average amplitude of saccades; maximum amplitude of saccades; total amplitude of saccades; saccade count/s; fixation duration; fixation dispersion) were investigated . The result was that the blink count could not be used as a marker for mind-wandering during learning video lectures among them (oculomotor data and eye movements), unlike previous literatures. Based on the results of this study, they identified elements that can be used as mind-wandering markers while learning from video lectures that are similar to real classes, among the oculomotor data and eye movement mentioned in previous literatures. Also, they found that most participants focused on past thoughts and felt unpleasant after experiencing mind-wandering through interview analysis.

**Related work:** First, participants were trained by researchers to distinguish attention from MW and from attention such as T.R.T(Task Related Thought) and on-task attention. Afterwards, they practiced the self-caught method of reporting MW immediately after recognizing their experience with MW. The participants' distance from the screen and the height of their chairs were adjusted through pre-testing so that their eye movements could easily be tracked as they took positions that would be comfortable for a 30-minute duration. The measurement was conducted in a slide room so that participants could watch the video lectures alone and without interruption. Precautions were provided for 20 seconds after the eye-tracker was adjusted through 9-point calibrations. Subsequently, the participants were asked to stare at the "□" in a picture that captured the first video lecture scene to measure the baseline of their pupil sizes. A green screen was shown for 30 seconds before and after the baseline screen for accurate measurement. Finally, the video was played and the students were asked to place their finger on the upper left corner of the mouse throughout the watching of the video to prepare for self-capture. The oculomotor data and eye movements of the participants were recorded by the eye-tracker during the test. This test process is illustrated in Figure 3.2.


**Figure 3.2: trial process with Tobii pro lab software**

**Article 3: Engagement detection in online learning: a review**

**Author:** M. Ali Akber Dewan , Mahbub Murshed and Fuhua Lin

**University:** School of Computing and Information Systems, Faculty of Science and Technology, Athabasca University, Athabasca, Alberta, Canada

**Abstract:** To provide personalized pedagogical support through interventions to online learners, it is important for online educators to detect their online learners' engagement status precisely and efficiently. This paper presents a review of the state of the art in engagement detection in the context of online learning. They classified the existing methods into three main categories automatic, semi-automatic and manual, considering the methods' dependencies on learners' participation. Methods in each category are then divided into subcategories based on the data types (e.g., audio, video, texts for learner log data etc.) they process for the engagement detection. In particular, the computer vision based methods in the automatic category that use facial expressions are examined in more details because they are found to be promising in the online learning environment. These methods are nonintrusive in nature, and the hardware and the software that these methods use to capture and analyze video data are cost-effective and easily achievable. Different techniques in the field of computer vision and machine learning are applied in these methods for the engagement detection. They also identified challenges of engagement detection and explore available datasets and performance metrics for engagement detection, and provide recommendations for the future to advance the technology of engagement detection for online education.

**Related work:** This article presented a generic framework (figure 3.3) for learner's perceived engagement detection using the computer vision based methods. The framework consisted with five different modules that include detection, feature extraction, tracking, classification, and decision. In a computer vision based engagement detection system, video streams were captured using a webcam or a surveillance camera, where the camera provides a particular view of learners participating in a learning activity. The system seeks to detect the region of interests (ROIs) (e.g., face, gestures, postures or eye) of the learners in the live video stream. Typically, engagement detection in such system is performed with a track and-classify approach. The system first performs segmentation to isolate the ROIs using a detection module in each frame. For each ROI, features are then extracted in a feature extraction module and selected into patterns to initiate tracking and classification.

A classification module is used to match input patterns against patterns extracted from training dataset and generates classification scores.

A tracking module is designed for tracking the movement or changes in the ROIs in consecutive frames and generates tracking trajectories. Finally, a decision module combines classification scores over trajectories to output a list of engagement levels of the learners in the input video stream.

They found that the most commonly used modalities in computer vision based methods are facial expressions, gestures and postures, and eye movement.

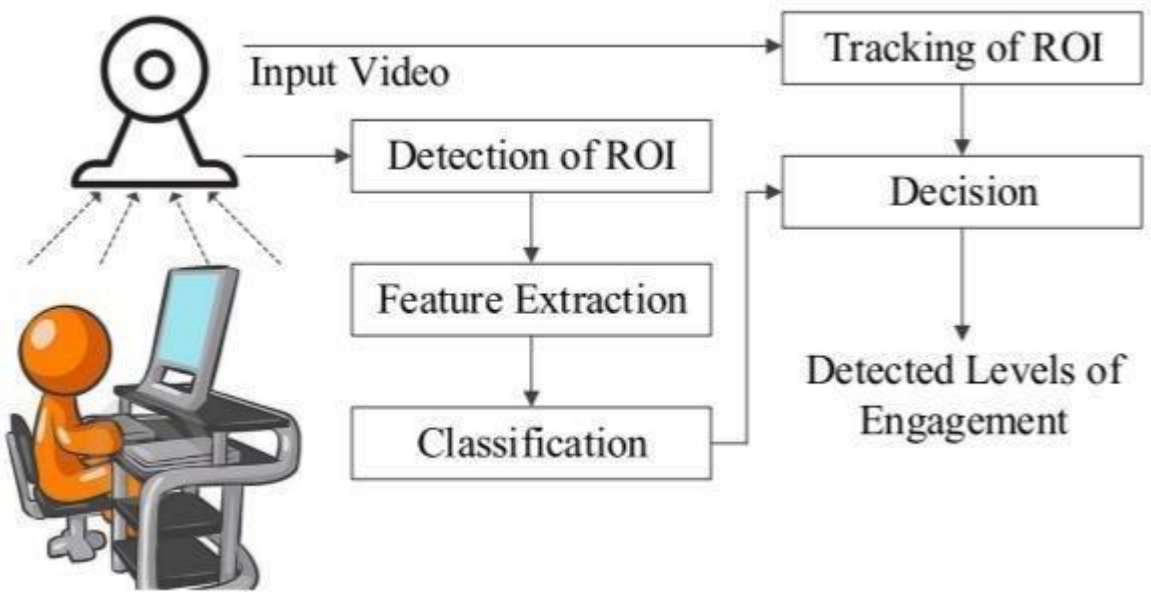



**Figure 3.3: A generic framework for computer vision based engagement detection system**

## Article 4: Learning analytics for IoE based educational model using deep learning techniques: architecture, challenges and applications

**Author:** Mohd Abdul Ahad* , Gautami Tripathi and Parul Agarwal

**University:** Department of Computer Science and Engineering, School of Engineering Sciences and Technology, Jamia Hamdard, New Delhi 110062, India

**Abstract:** The new generation teaching-learning pedagogy has created a complete paradigm shift wherein the teaching is no longer confined to giving the content knowledge, rather it fosters the "how, when and why" of applying this knowledge in real world scenarios. By exploiting the advantages of deep learning technology, this pedagogy can be further fine-tuned to develop a repertoire of teaching strategies. This paper presents a secured and agile architecture of an Internet of Everything (IoE) based. Educational Model and a Learning Analytics System (LAS) model using the concept of deep learning which can be used to gauge the degree of learning, retention and achievements of the learners and suggests improvements and corrective measures. The paper also puts forward the advantages, applications and challenges of using deep learning techniques for gaining insights from the data generated from the IoE devices within the educational domain for creating such learning analytics systems. Finally a feature wise comparison is provided between the proposed Learning Analytics (LA) based approach and conventional teaching-learning approach in terms of performance parameters like cognition, attention, retention and attainment of learners.

**Related work:** This paper presents an IoE based Educational model and discusses the applications, advantages and challenges of using deep learning techniques to develop a learning analytics system by effectively using the IoE big data for taking better and efficient decisions within the educational domain. Deep learning is a part of machine learning technique based on learning data representations. This learning can be unsupervised, supervised or semi-supervised. In other words we can say that, any deep learning model learns from the experience with minimal external interference. In a typical deep learning system, a computer model is created which is capable of performing the task of classification on the basis of input images, audios and videos. The training of these models is conducted using deep neural network architecture and huge datasets which are generally.


**Figure 3.4:** For the given input first, the relevant features are extracted, then these features are classified and machine learning algorithms are applied on them to perform the task of object classification

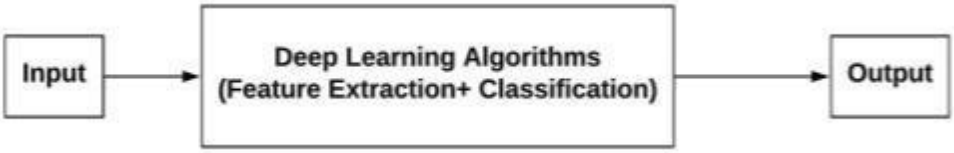



**Figure 3.5:** Deep Learning Approach - Here the feature extraction and classification task are done simultaneously

## Article 5: DAISEE: Dataset for Affective States in E-Learning Environments

**Author:** Abhay Gupta1 , Richik Jaiswal2 , Sagar Adhikari2 , Vineeth Balasubramanian2
**University:** Department of Computer Science, IIT
Hyderabad **Related work:**

### 3.5.1 Data Collection

**Data capture:** Data is captured using a full HD web camera (1920x1080, 30 fps, focal length 3.6mm, 78◦ field of view) mounted on a computer focusing on subjects watching e-learning videos. To simulate the e-learning environment, a custom application was created that presented a subject with 2 different videos (20 minutes total in length), one educational and one recreational to portray different learning environments. To model unconstrained settings, the subjects had the option to scroll through the videos. There are 95 subjects in the dataset belonging to the age group of 18-30, all of whom are currently enrolled students. In total, 30 hours of recordings were captured. The videos were captured in 5 different location settings:
(1) lab setting with high background clutter; (2) dorm room; (3) lab setting with minimal background; (4) white background; and (5) miscellaneous locations (random locations where the subject was found). Each of these settings has different illumination conditions, which were then categorized manually as low or high.

**Data Pre-Processing:** All recorded videos are divided into 10-second snippets and then individually processed for face detection using the standard Viola-Jones face detector. The snippets in which faces are detected across all frames are retained. The resulting dataset has 7338 video snippets, each video snippet being 10 seconds long across 95 subjects, 5 different locations, and 2 different illumination conditions.
**Hawthorne Effect, Subject Privacy, and Anonymity**: The Hawthorne effect, also referred to as the observer effect, is a type of reactivity in which individuals modify or improve an aspect of their behavior in response to their awareness of being observed. This is a critical aspect of such a data capture setting and it is highly probable that the subjects may adapt their behavior to suit the objectives of the experiment. To diminish the effects of such a circumstance, the subjects were recorded without their knowledge. This helped in limiting the Hawthorne effect. To account for the privacy interests of every subject, at the end of the data capture, they were informed of the recordings and their consent obtained to carry out research work.

### 3.5.2 Data Annotation

DAiSEE is created by tapping into the potential of crowd annotators. Crowd annotation brings in mass intelligence to interpret affective states that often can be subtle and prone to individual bias. Over the last few years, newer computer vision datasets are increasingly relying on wisdom-of-the-crowd for annotations, due to the large amounts of data and easy availability of annotators on crowd sourcing platforms. Although the annotators can be non-experts, it has been shown that repeated labeling of examples by multiple annotators can produce high-quality labels. Popular crowd sourcing platforms include Amazon's Mechanical Turk (AMT), CrowdFlower, LiveOps, InnoCentive, and Samasource1. These frameworks also provide interfaces for fast and reliable crowd annotation. In this work, we used CrowdFlower for the annotations.

**Class Labels:** Our dataset consists of labelling of four emotional states, viz., engagement, frustration, confusion and boredom. Recent work has shown that the six essential expressions: anger, disgust, fear, joy, sadness and surprise are not reliable in prolonged learning situations, such as classrooms and e-learning. Our choice of affective states such as engagement, frustration, confusion and boredom for this work is also supported by recent work in intelligent tutoring systems. For each of the affective states, we provide four labels:
(1) very low (2) low (3) high and (4) very high. This was obtained by conducting empirical studies with labels of other levels, including 5 and 3. In case of 5-scale, our labels included strongly positive, positive, neutral, negative and strongly negative; and the 3-scale study included positive, neutral and negative. Including neutral in the scales gave equivocal results as the annotators vote neutral when there is ambiguity. Hence, we chose a 4-scale/2-scale labelling strategy. A 4-scale is chosen over the 2-scale to increase the richness of the information of the labels offered and to also help learn the subtle differences in affective states of users in learning environments that extend beyond the dataset.

### 3.5.3 Salient features of dataset

Every video snippet in DAiSEE is labelled with four attributes, engagement, frustration, confusion and boredom. These attributes provide rich information about the learning experience of the subject, and can allow personalization of content as well as feedback for course, experience and environment design. The dataset takes into account the nature of different subjects across e-learning platforms at different locations. User affective states are captured at diverse locations which have significant background noise and clutter with people often walking around in the background and sometimes interacting with the subject. An important aspect of typical e-learning environments is lighting conditions for different subjects, and we have captured videos both in bright and dark settings to reproduce this artifact in the dataset. All these features demonstrate the resemblance of DAiSEE to videos captured in real-world e-learning environments.

**Figure 3.6: Examples of video snippets from dataset**

# CHAPTER-4

# 4. SYSTEM DESIGN

A good system design is to organise the program modules in such a way that are easy to develop and change. Structured design techniques help developers to deal with the size and complexity of programs.

The strategy used in attention span detection in instructor led online sessions is Bottom-Up approach. This design starts with the lowest level components and subsystems. By using these components, the next immediate higher level components and subsystems are composed or created. The process is continued till all the components and subsystems are composed into a single component, which is considered as the complete system. The amount of abstraction grows high as the design moves to more high levels.

## UML

The Unified Modelling Language is a standard language for writing software blueprints. It allows the software engineer to express an analysis model using the modelling notation that is governed by a set of syntactic, semantic and pragmatic rules. A UML system is represented using five different views that describe the system from distinctly different perspective. Each view is defined by a set of diagram, which is as follows:

1. User Model View: This view represents the system from the users' perspective. The analysis representation describes a usage scenario from the end-users' perspective.

2. Structural Model View: In this model, the data and functionality are arrived from inside the system. This model view models the static structures.

3. Behavioural Model View: It represents the dynamic of behavioural as parts of the system, depicting he interactions of collection between various structural elements described in the user model and structural model view.

4. Implementation Model View: In this view, the structural and behavioural as parts of the system are represented as they are to be built.

5. Environmental Model View: In this view, the structural and behavioural aspects of the environment in which the system is to be implemented are represented.

### 4.1 USE CASE DIAGRAM:

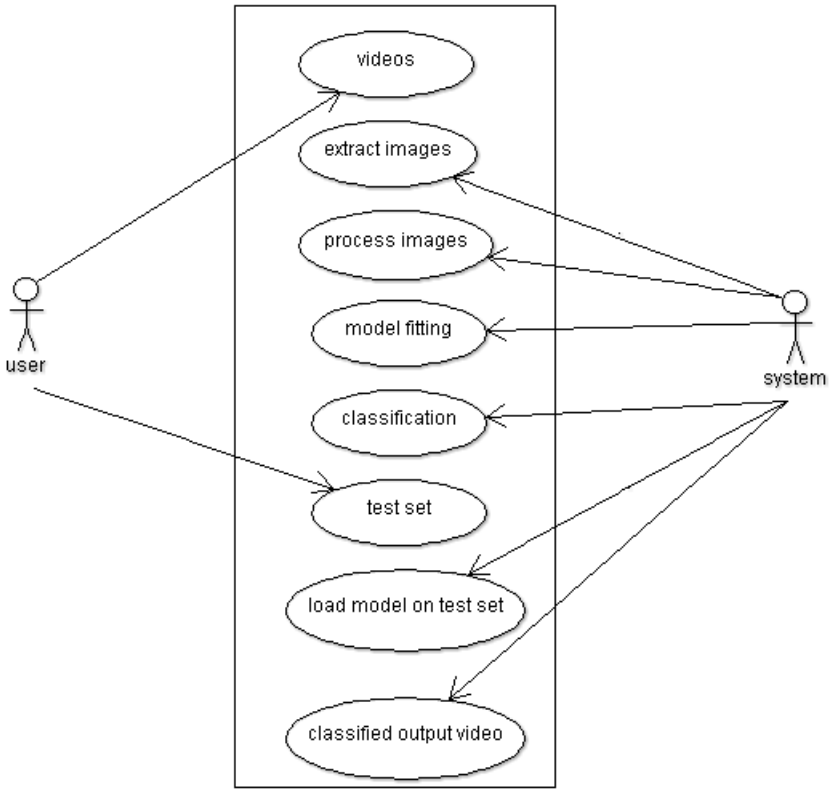



**Figure 4.1: use case diagram**

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses.

The purpose of use case diagrams can be said to be as follows −

- Used to gather the requirements of a system.
- Used to get an outside view of a system.
- Identify the external and internal factors influencing the system.
- Shows the interactions among the requirements are actors.

## 4.2 CLASS DIAGRAM:

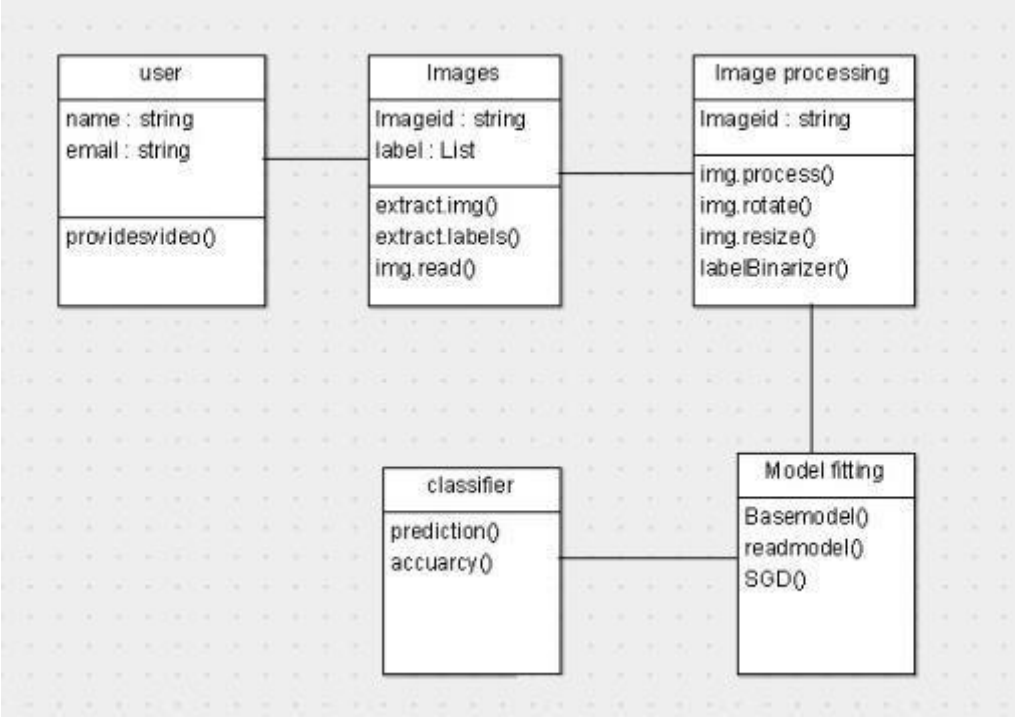



**Figure 4.2: class diagram**

The class diagram depicts a static view of an application. It represents the types of objects residing in the system and the relationships between them. A class consists of its objects, and also it may inherit from other classes. A class diagram is used to visualize, describe, document various different aspects of the system, and also construct executable software code. It shows the attributes, classes, functions, and relationships to give an overview of the software system. It constitutes class names, attributes, and functions in a separate compartment that helps in software development. Since it is a collection of classes, interfaces, associations, collaborations, and constraints, it is termed as a structural diagram. The class diagram is made up of three sections:

- Upper Section: The upper section encompasses the name of the class.
- Middle Section: The middle section constitutes the attributes, which describe the quality of the class.
- Lower Section: The lower section contains methods or operations.

## 4.3 SEQUENCE DIAGRAM:

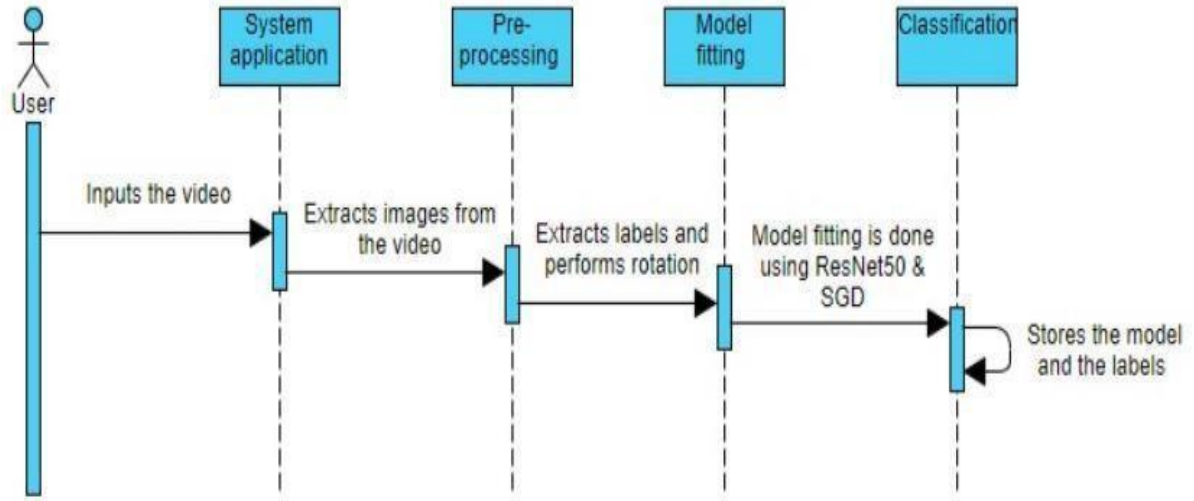



**Figure 4.3.1: sequence diagram for training**

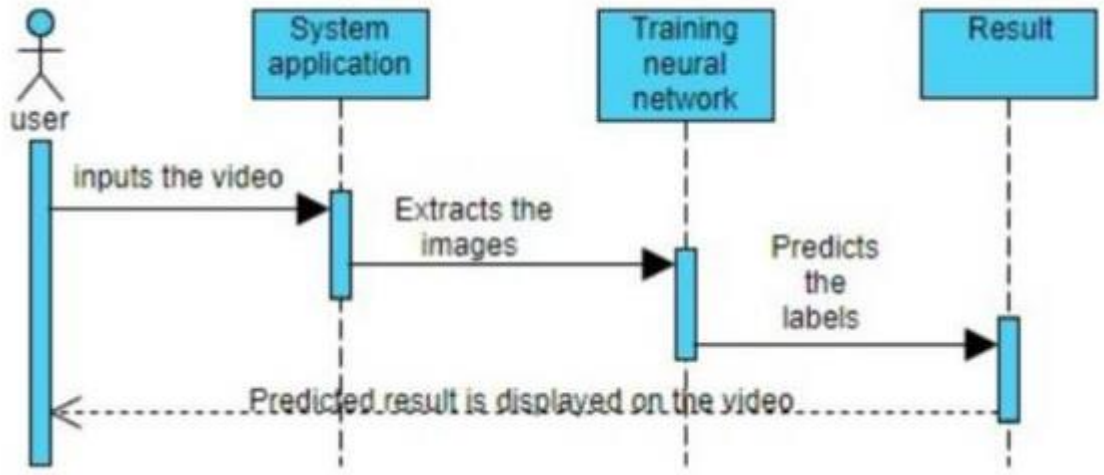



**Figure 4.3.2: sequence diagram for testing**

## Description:

Sequence diagrams describe interactions among classes in terms of an exchange of messages over time. They're also called event diagrams. A sequence diagram is a good way to visualize and validate various runtime scenarios. These can help to predict how a system will behave and to discover responsibilities a class may need to have in the process of modelling a new system.

The aim of a sequence diagram is to define event sequences, which would have a desired outcome. The focus is more on the order in which messages occur than on the message . However, the majority of sequence diagrams will communicate what messages are sent and the order in which they tend to occur.

## Sequence Diagram Notations:

**Actors –** An actor in a UML diagram represents a type of role where it interacts with the system and its objects.

**Lifelines –** A lifeline is a named element which depicts an individual participant in a sequence diagram. So basically each instance in a sequence diagram is represented by a lifeline. Lifeline elements are located at the top in a sequence diagram.

The difference between a lifeline and an actor is that a lifeline always portrays an object internal to the system whereas actors are used to depict objects external to the system.

**Messages –** Communication between objects is depicted using messages. The messages appear in a sequential order on the lifeline. We represent messages using arrows.

Lifelines and messages form the core of a sequence diagram.

Messages can be broadly classified into the following categories:

**Synchronous messages –** A synchronous message waits for a reply before the interaction can move forward. The sender waits until the receiver has completed the processing of the message.

**Asynchronous Messages –** An asynchronous message does not wait for a reply from the receiver. The interaction moves forward irrespective of the receiver processing the previous message or not.

## 4.4 ACTIVITY DIAGRAM:

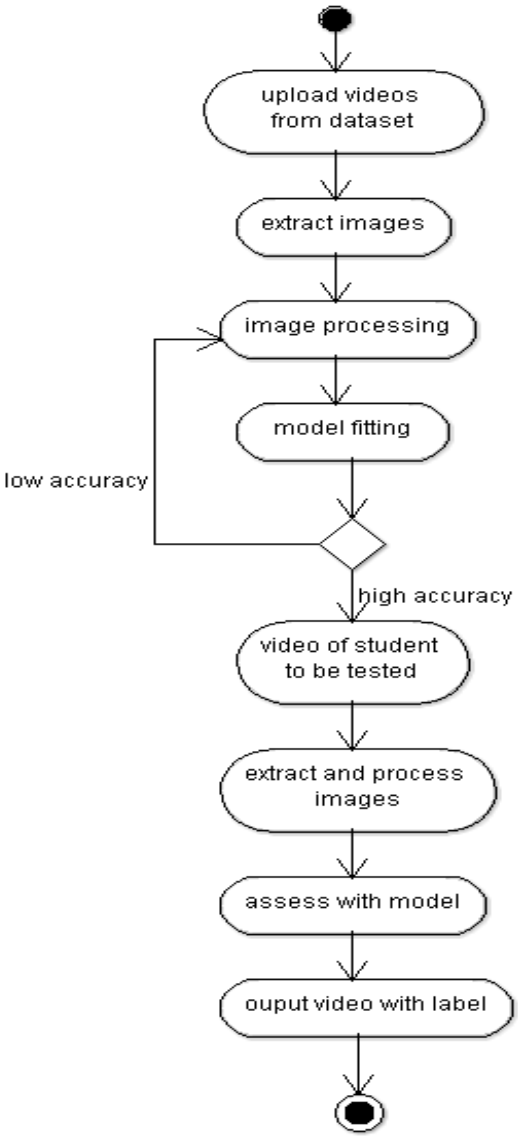



**Figure 4.4: Activity diagram**

Activity Diagram is used to illustrate the flow of control in a system and refer to the steps involved in the execution of a use case. Sequential and concurrent activities are modelled using activity diagrams. An activity diagram focuses on condition of flow and the sequence in which it happens. An activity diagram is a behavioral diagram that is it depicts the behavior of a system. It portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed.

## 4.5 COMPONENT DIAGRAM:

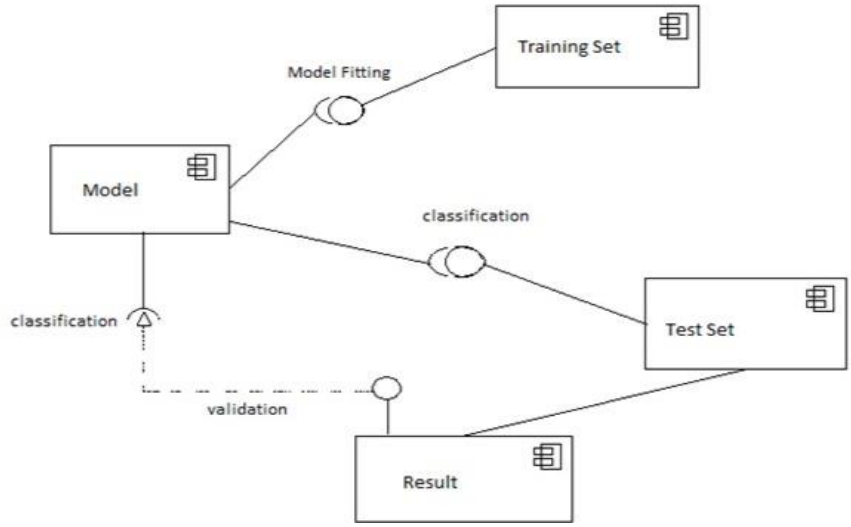



**Figure 4.5 component diagram**

A component diagram depicts how components are wired together to form larger components or software systems. They are used to illustrate the structure of arbitrarily complex systems. A component diagram allows verification that a system's required functionality is acceptable. These diagrams are also used as a communication tool between the developer and stakeholders of the system. Programmers and developers use the diagrams to formalize a roadmap for the implementation, allowing for better decision-making about task assignment or needed skill improvements

**Component Diagram Notations**

**Component**

A component is a logical unit block of the system, a slightly higher abstraction than classes. It is represented as a rectangle with a smaller rectangle in the upper right corner.

**Interface**

An interface describes a group of operations used or created by components.

**Dependencies**

Draw dependencies among components using dashed arrows.

**Port**

Ports are represented using a square along the edge of the system or a component.

# CHAPTER-5

# 5. IMPLEMENTATION

## 5.1 Frameworks Needed

### 5.1.1 Keras

Keras is one of the leading high-level neural networks APIs. It is written in Python and supports multiple back-end neural network computation engines. It was created to be user friendly, modular, easy to extend, and to work with Python. The API was “designed for human beings, not machines,” and “follows best practices for reducing cognitive load.” Neural layers, cost functions, optimizers, initialization schemes, activation functions, and regularization schemes are all standalone modules that we can combine to create new models. New modules are simple to add, as new classes and functions. Models are defined in Python code, not separate model configuration files.

The biggest reasons to use Keras stem from its guiding principles, primarily the one about being user friendly. Beyond ease of learning and ease of model building, Keras offers the advantages of broad adoption, support for a wide range of production deployment options, integration with at least five back-end engines (TensorFlow, CNTK, Theano, MXNet, and PlaidML), and strong support for multiple GPUs and distributed training.

### 5.1.2 TensorFlow

TensorFlow eases the process of acquiring data, training models, serving predictions, and refining future results. TensorFlow is an open source library for numerical computation and large-scale machine learning. TensorFlow bundles together a slew of machine learning and deep learning (aka neural networking) models and algorithms and makes them useful by way of a common metaphor. TensorFlow allows developers to create dataflowgraphs—structures that describe how data moves through a graph, or a series of processing nodes. Each node in the graph represents a mathematical operation, and each connection or edge between nodes is a multidimensional data array, or tensor. TensorFlow applications can be run on most any target that’s convenient: a local machine, a cluster in the cloud, iOS and Android devices, CPUs or GPUs. The resulting models created by TensorFlow, though, can be deployed on most any device where they will be used to serve predictions.

## 5.2 Libraries used

**Pandas:** Pandas is the most popular python library that is used for data analysis. It provides highly optimized performance with back-end source code is purely written in C or Python. Pandas make importing, analyzing, and visualizing data much easier. It is built on packages like NumPy and matplotlib to give you a single, convenient, place to do most of your data analysis and visualization work.
Command used to install pandas - pip install pandas

**Scikit-learn:** Scikit-learn is an open source Python library that implements a range of machine learning, pre-processing, cross-validation and visualization algorithms using a unified interface. It provides simple and efficient tools for data mining and data analysis. Scikit-learn is built on the top of NumPy, SciPy, and matplotlib libraries. Scikit-learn requires NumPy and SciPy as its dependencies.
Command used to install scikit-learn - pip install -u scikit-learn

**Numpy:** NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python. Besides its obvious scientific uses, Numpy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined using Numpy which allows NumPy to seamlessly and speedily integrate with a wide variety of databases.
Command used to install numpy - pip install numpy

**Matplotlib:** Matplotlib is a visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack. One of the greatest benefits of visualization is that it allows us visual access to huge amounts of data in easily digestible visuals. Matplotlib consists of several plots like line, bar, scatter, histogram etc. Plots helps to understand trends, patterns, and to make correlations. They’re typically instruments for reasoning about quantitative information.
Command used to install matplotlib - python -mpip install -U matplotlib

**Keras:** Keras is an open-source neural-network library written in Python. It is capable of running on top of TensorFlow, Microsoft Cognitive Toolkit, R, Theano, or PlaidML Keras is a powerful and easy-to-use free open source Python library for developing and evaluating deep learning models. It wraps the efficient numerical computation libraries Theano and TensorFlow and allows you to define and train neural network models in just a few lines of code.
Command used to install Keras - pip install Keras

**OpenCV:** Open source computer vision is a library of programming functions mainly aimed at real-time computer vision. OpenCV supports some models from deep learning frameworks like TensorFlow, Torch, PyTorch after converting to an ONNX model and Caffe according to a defined list of supported layers.
Command used to install opencv - pip install opencv-python

**Imutils:** Imutils are a series of convenience functions to make basic image processing functions such as translation, rotation, resizing, skeletonization, and displaying matplotlib images easier with OpenCV.
Command used to install imutils - pip install imutils

**OS:** The OS module in python provides functions for interacting with the operating system. OS, comes under Python's standard utility modules. This module provides a portable way of using operating system dependent functionality. The *os* and *os.path* modules include many functions to interact with the file system.

**Glob:** The glob module is used to retrieve files/pathnames matching a specified pattern. The pattern rules of glob follow standard Unix path expansion rules. With glob, we can also use wildcards ("*, ?, [ranges]) apart from exact string search to make path retrieval more simple and convenient.

**Pickle:** Python pickle module is used for serializing and de-serializing a python object structure. Any object in Python can be pickled so that it can be saved on disk. Pickle "serializes" the object first before writing it to file. Pickling is a way to convert a python object (list, dict, etc.) into a character stream.

**CSV:** Comma Separated Values is a simple file format used to store tabular data, such as a spreadsheet or database. A CSV file stores tabular data (numbers and text) in plain text. Each line of the file is a data record. Each record consists of one or more fields, separated by commas. The use of the comma as a field separator is the source of the name for this file format. For working CSV files in python, there is an inbuilt module called csv.

**Argparse:** Argparse is a complete argument processing library. The argparse module makes it easy to write user friendly command-line interfaces. The program defines what arguments it requires, and argparse will figure out how to parse those out of sysargv. The argparse module also automatically generates help and usage messages and issues errors when users give the program invalid arguments.

**Collections :** Collections is a built-in Python module that implements specialized container datatypes such as queues and dequeues providing alternatives to Python's general purpose built-in containers such as dict, list, set, and tuple.

## 5.3 Classification Algorithm

### 5.3.1 Convolution Neural Network

The convolutional neural network (CNN) is a class of deep learning neural networks. They are most commonly used to analyze visual imagery and are frequently working behind the scenes in image classification. CNNs are inspired by biological processes. They are based on research done by Hubel and Wiesel in the 60's regarding vision in cats and monkeys. The pattern of connectivity in a CNN comes from their research regarding the organization of the visual cortex.

CNNs have an input layer, and output layer, and hidden layers. The hidden layers usually consist of convolutional layers, ReLU layers, pooling layers, and fully connected layers. Convolutional layers apply a convolution operation to the input. This passes the information on to the next layer. Pooling combines the outputs of clusters of neurons into a single neuron in the next layer. Fully connected layers connect every neuron in one layer to every neuron in the next layer. In a convolutional layer, neurons only receive input from a subarea of the previous layer. In a fully connected layer, each neuron receives input from every element of the previous layer.

A CNN works by extracting features from images. This eliminates the need for manual feature extraction. The features are not trained. They are learned while the network trains on a set of images. This makes deep learning models extremely accurate for computer vision tasks. CNNs learn feature detection through tens or hundreds of hidden layers. Each layer increases the complexity of the learned features.

A CNN

- starts with an input image
- Applies many different filters to it to create a feature map
- applies a ReLU function to increase non-linearity
- applies a pooling layer to each feature map
- flattens the pooled images into one long vector.
- inputs the vector into a fully connected artificial neural network.
- processes the features through the network. The final fully connected layer provides the “voting” of the classes that we are after.
- trains through forward propagation and back propagation for many, many epochs. This repeats until we have a well-defined neural network with trained weights and feature detectors.

## 5.4 Pseudo code

### 5.4.1 Creation of images

1) Collect the videos of all the students to be trained.
2) Loop over all videos and convert each video into images.
3) Extract the labels from each image.

### 5.4.2 Training

1) Loop over all the images extracted from videos
2) Pre-process the images
3) Perform one hot encoding on labels
4) Load ResNet50 pre-trained with imageNet weights as the baseModel
5) Create a new headModel using the base model
6) Compile the model with Stochastic Gradient Descent
7) Save the model and label binarizer
8) Evaluate the network

### 5.4.3 Testing

1) Load the model and label binarizer
2) Provide the path of student's video to be tested
3) Read frames from the video stream and pre-process each frame
4) Make predictions on the current frame
5) Find the label with with the largest corresponding probability across the average predictions
6) Draw the prediction on the output frame

## Import necessary packages

```
from keras.preprocessing.image import ImageDataGenerator
from keras.layers.pooling import AveragePooling2D
from keras.applications import ResNet50
from keras.layers.core import Dropout
from keras.layers.core import Flatten
from keras.layers.core import Dense
from keras.layers import Input
from keras.models import Model
from keras.optimizers import SGD
from sklearn.preprocessing import LabelBinarizer
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
from imutils import paths
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import argparse
import pickle
import cv2
import os
```

## 5.5 Code snippets

### 5.5.1 Creation of images

**Description:** The below code reads a csv file which contains the labels of the students to be trained and maps each label to the corresponding video of the student.

```
import csv
fields=[]
rows=[]
with open('C:\\Users\\devishriarugula\\Desktop\\Major Project\\TrainLabels.csv') as csvfile:
    read = csv.reader(csvfile)
    fields=next(read)
    for row in read:
        rows.append(row)

for i in rows[:]:
    clipid.append(i[0])
    c=1
    attention=""
    for j in i[1:]:
        if int(j)>0:
            if len(attention)>0:
                attention=attention+"-"+fields[c]
            else:
                attention=fields[c]
        c=c+1
    attentive.append(attention)

#creating a dictionary for video and labels
dict={}
for i in range(5358):
    dict[clipid[i]]=attentive[i]
```

**Description:** The below code takes videos of students and converts each video of 10 seconds into 10 jpg frames and stores them in a directory. This code also creates a csv file using pandas which contains image names and their respective labels. Glob() function is used to extract files in a recursive manner. VideoCapture() is used to capture videos.

```
files=glob.glob("C:\\Users\\devishriarugula\\Desktop\\Major Project\\200050/*/*",recursive=True)
count=0
for myfile in files:
    fname=os.path.split(myfile)
    for i in dict.keys():
        if(i==fname[1]):
            frame1=dict[i]
    cam=cv2.VideoCapture(myfile)
    while(True):
        cam.set(cv2.CAP_PROP_POS_MSEC,(cf*1000))
        ret,frame=cam.read()
        if ret:
            name='./stud1/'+'frame'+str(count)+frame1+'.jpg'
            image.append(name)
            labels.append(frame1)
            cv2.imwrite(name, frame)
            cf=cf+1
            count=count+1

cam.release()
cv2.destroyAllWindows()
data = pd.DataFrame()
data['image']=image
data['labels']=labels
data.to_csv('labels.csv',header=True, index=False)
```

### 5.5.2 Training

**Description:** The below code loops over all the image paths of the frames in the directory created by frames.py and load and preprocess an image. Preprocessing includes swapping color channels for OpenCV to Keras compatibility and resizing to 224×224px. Y stores the extracted column 'labels' from the csv file train created in labels.py. Data and y are converted into numpy arrays. One-hot encoding of labels is done which is a way of marking an active class label via binary array elements.

```
imagePaths = list(paths.list_images("team4/student1"))
for imagePath in imagePaths:
    image = cv2.imread(imagePath)
    image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
    image = cv2.resize(image, (224, 224))
    data.append(image)
data = np.array(data)
train = pd.read_csv('labels.csv')
y=train['labels']
labels=np.array(y)
lb = LabelBinarizer()
labels = lb.fit_transform(labels)
(trainX, testX, trainY, testY) = train_test_split(data, labels,test_size=0.25)
```

**Description:** The below code initializes two data augmentation objects — one for training and one for validation. The trainAug object performs random rotations, zooms, shifts, shears, and flips on our data. No augmentation will be conducted for validation data (valAug ), but mean subtraction is performed. The mean pixel value is set for trainAug and valAug so that mean subtraction will be conducted as images are generated during training/evaluation.

```
trainAug = ImageDataGenerator(
    rotation_range=30,
    zoom_range=0.15,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.15,
    horizontal_flip=True,
    fill_mode="nearest")
valAug = ImageDataGenerator()
mean = np.array([123.68, 116.779, 103.939], dtype="float32")
trainAug.mean = mean
valAug.mean = mean
```

**Description:** Import a model named ResNet50 from keras.applications. ResNet-50 is a convolutional neural network that is trained on more than a million images from the ImageNet database. The network is 50 layers deep and can classify images into thousand object categories, such as keyboard, mouse, pencil, and many animals. Once the model is generated we will be able to display the accuracy and the classification report.

The below code loads ResNet50 pre-trained with ImageNet weights while chopping the head of the network off and assemble a new headModel and suture it onto the baseModel. Then the code compiles our model with the Stochastic Gradient Descent (SGD ) optimizer with an initial learning rate of 1e-4 and learning rate decay. We use "categorical_crossentropy" loss for training with multiple classes. A call to the fit_generator function on our model trains our network with data augmentation and mean subtraction.

```
baseModel = ResNet50(weights="imagenet", include_top=False,
                     input_tensor=Input(shape=(224, 224, 3)))
headModel = baseModel.output
headModel = AveragePooling2D(pool_size=(7, 7))(headModel)
headModel = Flatten(name="flatten")(headModel)
headModel = Dense(512, activation="relu")(headModel)
headModel = Dropout(0.5)(headModel)
headModel = Dense(len(lb.classes_), activation="softmax")(headModel)
model = Model(inputs=baseModel.input, outputs=headModel)

import argparse
ap = argparse.ArgumentParser()

ap.add_argument("-e", "--epochs", type=int, default=25,
                help="# of epochs to train our network for")
args = vars(ap.parse_args())

print("[INFO] compiling model...")
opt = SGD(lr=1e-4, momentum=0.9, decay=1e-4 / args["epochs"])
model.compile(loss="categorical_crossentropy", optimizer=opt,metrics=["accuracy"])

print("[INFO] training head...")
H = model.fit_generator(trainAug.flow(trainX, trainY, batch_size=32),
                        steps_per_epoch=len(trainX) // 32,
                        validation_data=valAug.flow(testX, testY),
                        validation_steps=len(testX) // 32,epochs=args["epochs"])
```

The below code evaluates our network on the testing set and prints a classification report, saves our fine-tuned keras model then serializes and stores label binarizer in python's pickle format.

```
print("[INFO] evaluating network...")
predictions = model.predict(testX, batch_size=32)
print(classification_report(testY.argmax(axis=1),
                            predictions.argmax(axis=1),
                            target_names=lb.classes_))

print("[INFO] serializing network...")
model.save('Model1')
f = open('label-bin1', "wb")
f.write(pickle.dumps(lb))
f.close()
```

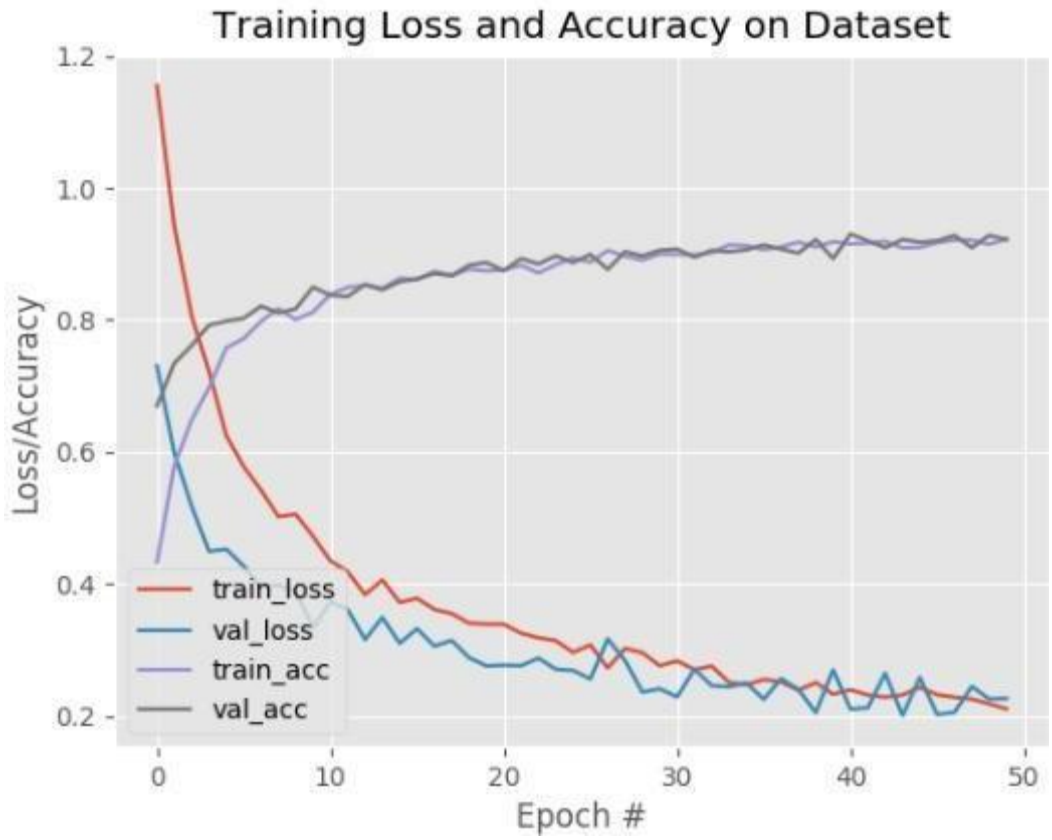



**Figure 5.5.1: video classification with keras accuracy/loss training history plot**

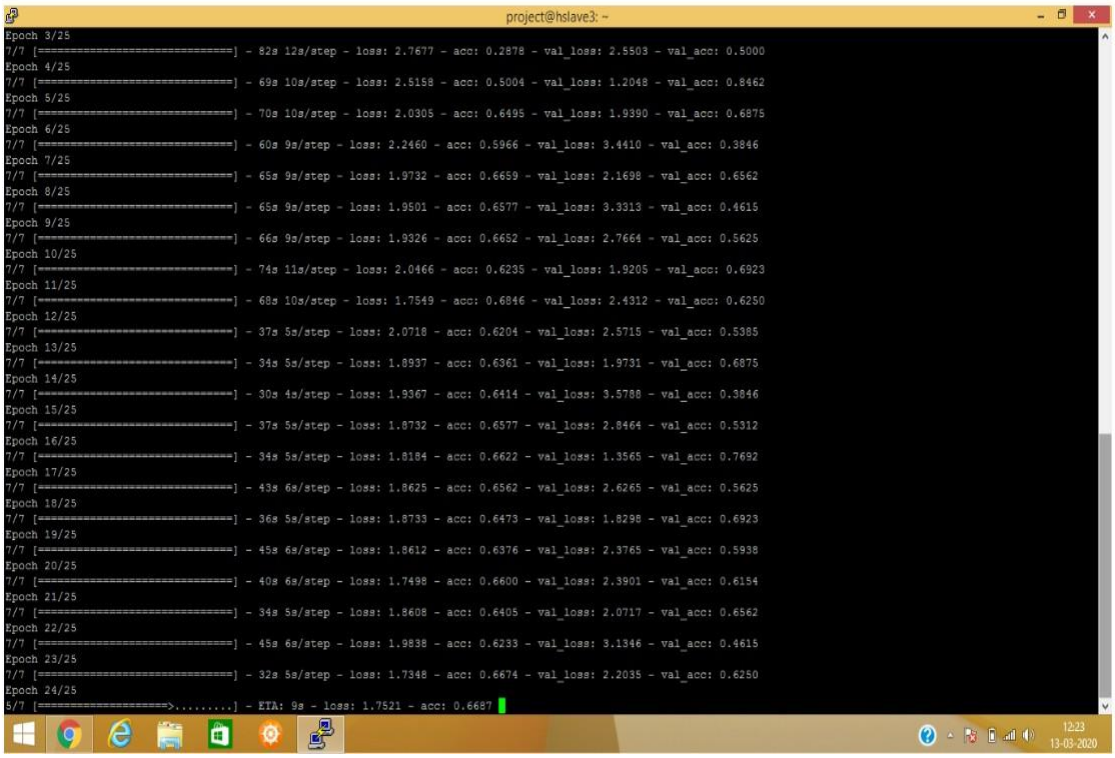
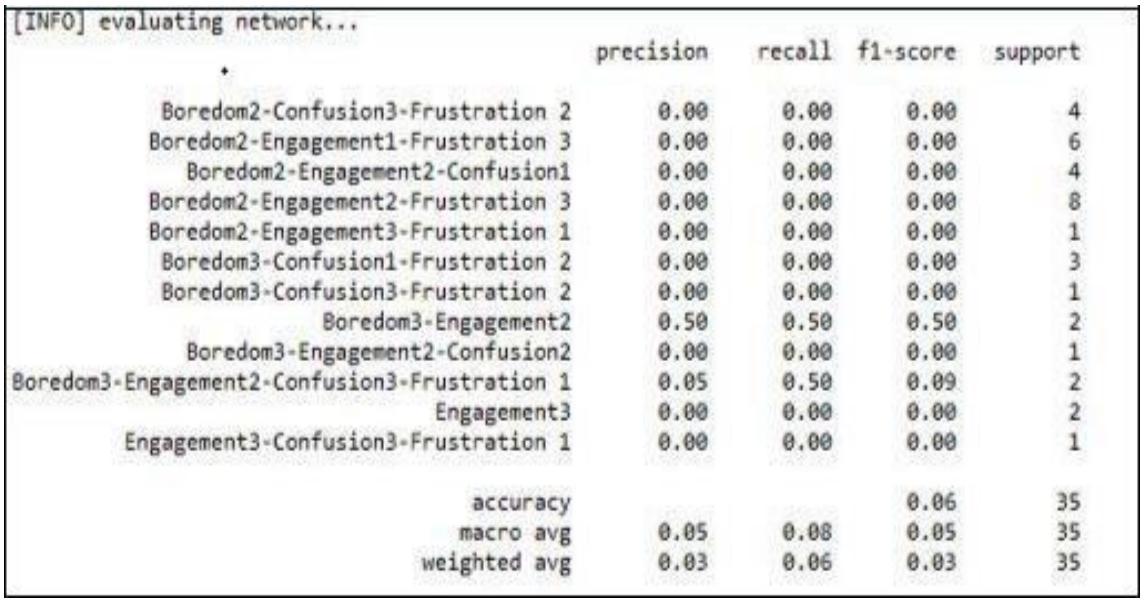



**Figure 5.5.2: increasing accuracy while training**



[INFO] evaluating network...

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| Boredom2-Confusion3-Frustration 2 | 0.00 | 0.00 | 0.00 | 4 |
| Boredom2-Engagement1-Frustration 3 | 0.00 | 0.00 | 0.00 | 6 |
| Boredom2-Engagement2-Confusion1 | 0.00 | 0.00 | 0.00 | 4 |
| Boredom2-Engagement2-Frustration 3 | 0.00 | 0.00 | 0.00 | 8 |
| Boredom2-Engagement3-Frustration 1 | 0.00 | 0.00 | 0.00 | 1 |
| Boredom3-Confusion1-Frustration 2 | 0.00 | 0.00 | 0.00 | 3 |
| Boredom3-Confusion3-Frustration 2 | 0.00 | 0.00 | 0.00 | 1 |
| Boredom3-Engagement2 | 0.50 | 0.50 | 0.50 | 2 |
| Boredom3-Engagement2-Confusion2 | 0.00 | 0.00 | 0.00 | 1 |
| Boredom3-Engagement2-Confusion3-Frustration 1 | 0.05 | 0.50 | 0.09 | 2 |
| Engagement3 | 0.00 | 0.00 | 0.00 | 2 |
| Engagement3-Confusion3-Frustration 1 | 0.00 | 0.00 | 0.00 | 1 |
| accuracy | | | 0.06 | 35 |
| macro avg | 0.05 | 0.08 | 0.05 | 35 |
| weighted avg | 0.03 | 0.06 | 0.03 | 35 |

**Figure 5.5.3: confusion matrix**

### 5.5.3 Testing

**Description:** The below code loads our model and label binarizer, then sets our mean subtraction value. Dequeue is initialized to maximum length of default value of 128.

```
ap.add_argument("-s", "--size", type=int, default=128,
                help="size of queue for averaging")
args = vars(ap.parse_args())

print("[INFO] loading model and label binarizer...")
model = load_model('Model1')
lb = pickle.loads(open('manisha1', "rb").read())

mean = np.array([123.68, 116.779, 103.939][::1], dtype="float32")
Q = deque(maxlen=args["size"])
```

**Description:** The below code initializes cv2.VideoCapture object and begin looping over video frames. Each frame is preprocessed using the same steps as training script. Predictions are made on that current frame and the prediction results are added to dequeue(Q). Then prediction averaging is performed over Q history resulting in a class label for the rolling average. Broken down, these lines find the label with the largest corresponding probability across the average predictions

```
vs = cv2.VideoCapture('C:\\Users\\Desktop\\2000501020\\2000501020.avi')
writer = None
(W, H) = (None, None)
currentframe=0
while True:
    vs.set(cv2.CAP_PROP_POS_MSEC,(currentframe*1000))
    (grabbed, frame) = vs.read()
    print(grabbed)
    if not grabbed:
        break
    if W is None or H is None:
        (H, W) = frame.shape[:2]
    currentframe=currentframe+1
    output = frame.copy()
    frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    frame = cv2.resize(frame, (224, 224)).astype("float32")
    frame -= mean
    preds = model.predict(np.expand_dims(frame, axis=0))[0]
    Q.append(preds)
    results = np.array(Q).mean(axis=0)
    i = np.argmax(results)
    label = lb.classes_[i]
```

**Description:** Now that we have our resulting label, we annotate the output frame and write it to disk.

The below code draws the prediction on the output frame, initializes the writer if necessary and the output file is written to the file. The output is also displayed on the screen until the "q" key is pressed or until the end of the video file is reached. Finally a cleanup action is performed.

```
text = "activity: {}".format(label)
    cv2.putText(output, text, (35, 50), cv2.FONT_HERSHEY_SIMPLEX,1.25, (0, 255, 0), 5)
    if writer is None:
        fourcc = cv2.VideoWriter_fourcc(*"MJPG")
        writer = cv2.VideoWriter("output1.", fourcc, 30,(W, H), True)
    writer.write(output)
    cv2.imshow("Output", output)
    key = cv2.waitKey(1) & 0xFF
    if key == ord("q"):
        break
print("[INFO] cleaning up...")
writer.release()
vs.release()
```

# CHAPTER-6

# 6. SYSTEM TESTING

Testing is the process of evaluating a system or its components with the intent to find whether it satisfies the specified requirements or not. Testing is executing a system in order to identify any gaps, errors, or missing requirements in contrary to the actual requirements. According to ANSI/IEEE 1059 standard, Testing can be defined as-A process of analyzing a software item to detect the difference between existing and required conditions (that is defects/errors/bugs) and to evaluate the features of the software item. It is one of the main stages of project development life cycle to provide our cessation utilize with information about the quality of the application. It can be either done manually or using automated tools.

Unit testing is performed in the development stage of Attention Span Detection in Instructor led Online Sessions to convert videos to images. In the implementation stage of the model, manual testing is done by considering videos of different students with various emotions. Browser compatibility testing is done on different web browsers and on different operating systems such as Windows, Linux. Lastly client side validation testing is performed successfully on our model.

## 6.1 UNIT TESTING

Unit testing is a level of software testing where individual units/components of software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output. In procedural programming, a unit may be an individual program, function, procedure, etc. In object-oriented programming, the smallest unit is a method, which may belong to a base/super class, abstract class or derived/child class. Unit testing is done during the development (coding) of an application.

Various errors encountered in the development stage of the model are:

### 6.1.1 Value error: Inconsistent number of samples

```
ValueError                                Traceback (most recent call last)
<ipython-input-2-95effc802604> in <module>()
     13 lb = LabelBinarizer()
     14 labels = lb.fit_transform(labels)
---> 15 (trainX, testX, trainY, testY) = train_test_split(data, labels,test_size=0.25)
     16 print(trainY)

~\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py in train_test_split(*arrays, **options)
   2029         test_size = 0.25
   2030
-> 2031     arrays = indexable(*arrays)
   2032
   2033     if shuffle is False:

~\Anaconda3\lib\site-packages\sklearn\utils\validation.py in indexable(*iterables)
    227         else:
    228             result.append(np.array(X))
--> 229     check_consistent_length(*result)
    230     return result
    231

~\Anaconda3\lib\site-packages\sklearn\utils\validation.py in check_consistent_length(*arrays)
    202     if len(uniques) > 1:
    203         raise ValueError("Found input variables with inconsistent numbers of"
--> 204                          " samples: %r" % [int(l) for l in lengths])
    205
    206

ValueError: Found input variables with inconsistent numbers of samples: [1100, 110]
```

**Figure 6.1.1: value error**

This error arises when X and Y do not have the same length i.e., X.shape[0] != Y.shape[0]. In our code the length of data is different from that of the labels. To resolve this error delete labels file, reassign the labels from csv file and remap them to each data frame respectively.

### 6.1.2 Module Not Found Error: keras

```
ModuleNotFoundError                       Traceback (most recent call last)
<ipython-input-1-d8cba512726f> in <module>()
----> 1 from keras.preprocessing.image import ImageDataGenerator
      2 from keras.layers.pooling import AveragePooling2D
      3 from keras.applications import ResNet50
      4 from keras.layers.core import Dropout
      5 from keras.layers.core import Flatten

ModuleNotFoundError: No module named 'keras'
```

**Figure 6.1.2: module not found error-keras**

This error arises when the specified module is not installed in the system. To resolve this error, we need to install a few dependencies like numpy, scikit-learn, tensorflow. Then we can install keras using the command – pip install keras.

### 6.1.3 Value error: Mix of label input types

```
ValueError                                Traceback (most recent call last)
<ipython-input-32-f347476bc78d> in <module>()
      1 a=[1,2,'a']
      2 lb = LabelBinarizer()
----> 3 labels = lb.fit_transform(a)

~\Anaconda3\lib\site-packages\sklearn\preprocessing\label.py in fit_transform(self, y)
    305             Shape will be [n_samples, 1] for binary problems.
    306         """
--> 307         return self.fit(y).transform(y)
    308
    309     def transform(self, y):

~\Anaconda3\lib\site-packages\sklearn\preprocessing\label.py in fit(self, y)
    282
    283         self.sparse_input_ = sp.issparse(y)
--> 284         self.classes_ = unique_labels(y)
    285         return self
    286

~\Anaconda3\lib\site-packages\sklearn\utils\multiclass.py in unique_labels(*ys)
    101     # Check that we don't mix string type with number type
    102     if (len(set(isinstance(label, string_types) for label in ys_labels)) > 1):
--> 103         raise ValueError("Mix of label input types (string and number)")
    104
    105     return np.array(sorted(ys_labels))

ValueError: Mix of label input types (string and number)
```

**Figure 6.1.3: value error**

This error arises when we try to mix numbers and strings, it's unclear whether we are mixing different types of classes, or if we are mixing continuous and non-continuous data. If the latter- you don't want the LabelBinarizer to run on the continuous data, and you should remove it, then re-add to the data later. If the former, you can convert the integers to strings.

### 6.1.4 Module not found error: cv2

```
Traceback (most recent call last):
  File "E:\data_struc\cv.py", line 1, in <module>
    import cv2
ModuleNotFoundError: No module named 'cv2'
```

**Figure 6.1.4 module not found error-cv2**

This error arises when the specified module is not installed in the system. To resolve this error install the specified model according to the python version. If cv2 is already installed and still the above error occurs the python version may not be compatible.

## 6.2 Integration testing:

Integration testing is a level of software testing where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in Integration Testing.

When the instructor wants to detect the attention of a student, the student's path is given as the input to the model thereby we obtain a video with the level of attention as a number prefixed with the label obtained.

## 6.3 Screenshots

Some of the screenshots of students whose attention span is detected are:

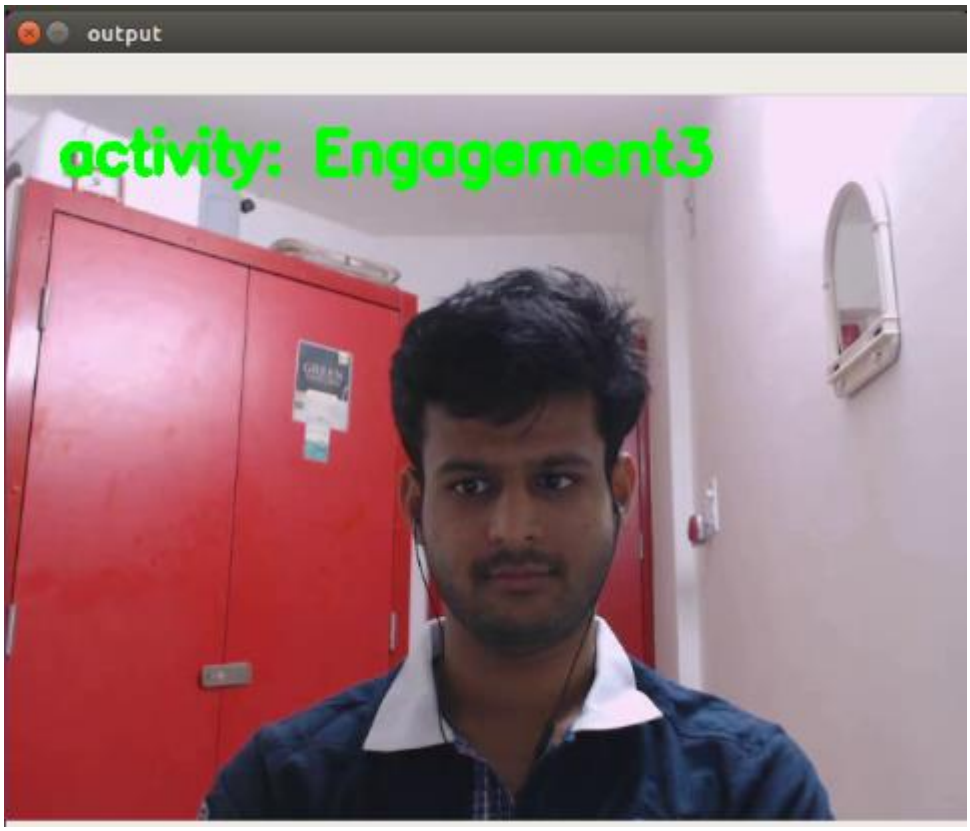



**Figure 6.3.1: student1**

The above figure is a screenshot from the output video of a student who is in 3rd level of engagement state that is he is very attentive towards the lecture he is listening.

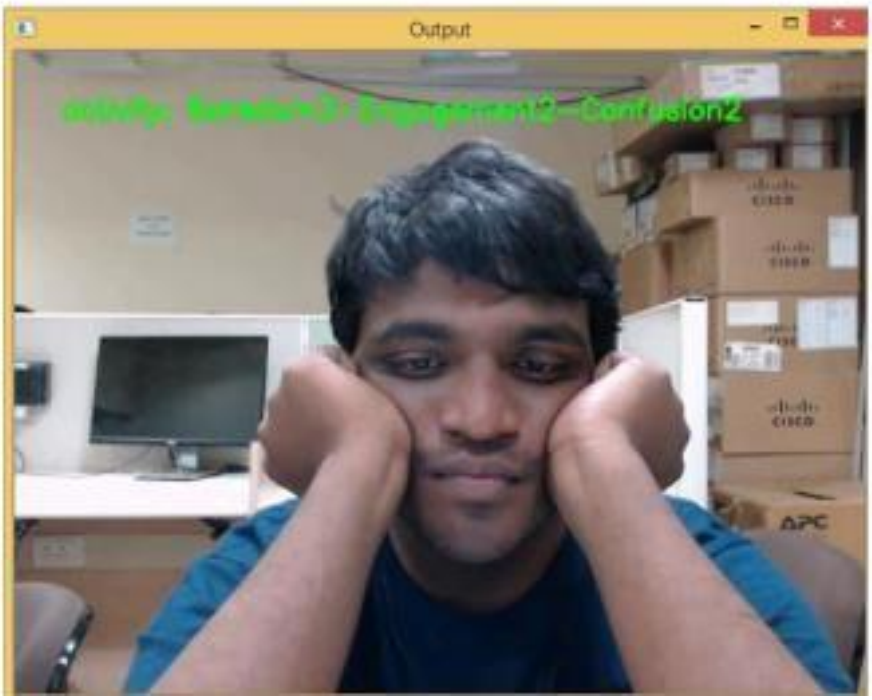



**Figure 6.3.2: student2**

The above figure is a screenshot from the output video of a student who is in 3rd level of boredom state, 2nd level of engagement state and also 2nd level of confusion state i.e he is bored and a little confused towards the lecture he is listening.

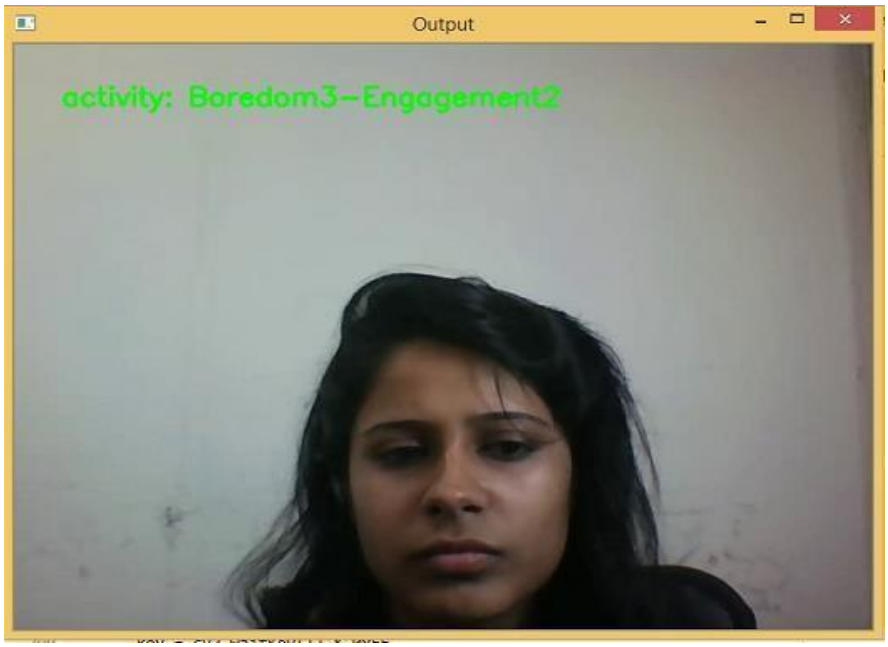



**Figure 6.3.3: student3**

The above figure is a screenshot from the output video of a student who is in 3rd level of boredom state and 2nd level of engagement state i.e she is bored while listening the lecture.


**Figure 6.3.4: student4**

The above figure is a screenshot from the output video of a student who is in 3rd level of engagement state, 3rd level of confusion state and 1st level of frustration state i.e he is confused and a little frustrated towards the lecture he is listening.

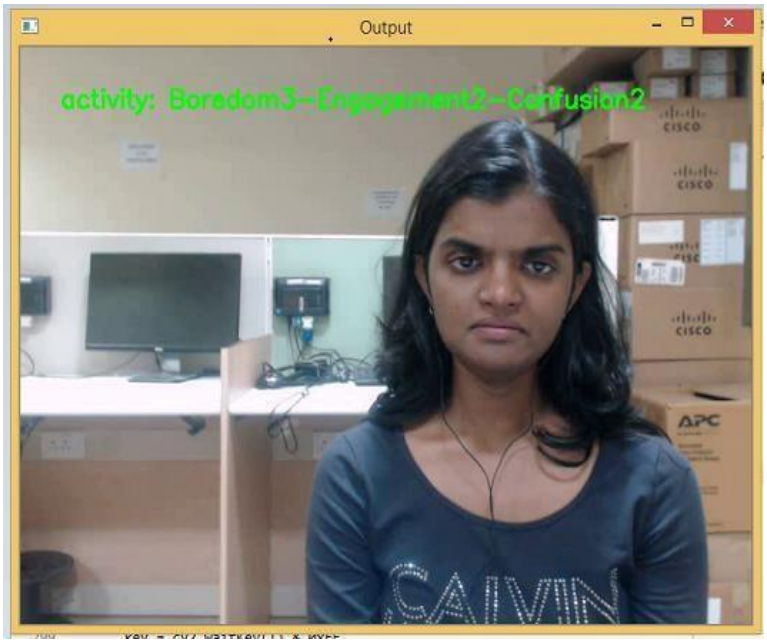



**Figure 6.3.5: student5**

The above figure is a screenshot from the output video of a student who is in 3rd level of boredom state, 2nd level of engagement state and also 2nd level of confusion state i.e she is bored and a little confused towards the lecture he is listening.

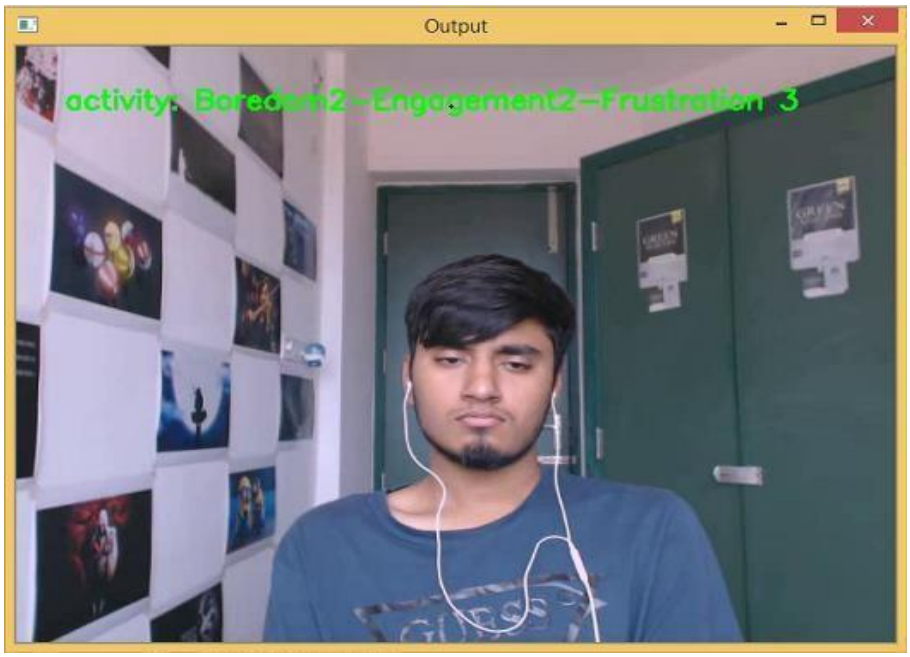



**Figure 6.3.6: student6**


**Figure 6.3.7: student7**

The above figures are screenshots from the output videos of the students who are in 2nd level of boredom state, 2nd level of engagement state and also 3rd level of frustration state i.e they are bored and frustrated while listening the lecture.


**Figure 6.3.8: student8**

The above figure is a screenshot from the output video of a student who is in 3rd level of boredom state and 2nd level of engagement state i.e she is bored while listening the lecture.

## System testing

System testing is a level of testing that validates the complete and fully integrated software product. The purpose of a system test is to evaluate the end-to-end system specifications. Usually, the software is only one element of a larger computer-based system. Ultimately, the software is interfaced with other software/hardware systems. System Testing is actually a series of different tests whose sole purpose is to exercise the full computer-based system.

## Acceptance testing

Acceptance testing is a level of software testing where a system is tested for acceptability. The purpose of this test is to evaluate the system's compliance with the business requirements and assess whether it is acceptable for delivery.

# CHAPTER - 7

# 7. CONCLUSION

Inter-personal human communication includes not only spoken languages but also non-verbal cues such as hand gestures and facial expressions which are used to express feelings and give feedback. Affective states such as engagement, frustration, confusion, and boredom are very important not only to express our emotions but also to provide important suggestions during social interactions such as level of interest, our desire to take a speaking turn or to provide continuous feedback on the understanding of the information conveyed. E-learning environments are one of the best examples for studying user affective states.

The model developed in this project monitors and measures the attention span of the learners in the instructor led online sessions by automatically recognizing students' affective states such as engagement, frustration, confusion and boredom frustration in e-learning environments.

The model takes a video stream as input, converts it into multiple images, processes them, extracts labels from images, fits the model using ResNet-50 as the base model and compiles it using Stochastic Gradient Descent. This model is saved and used repetitively to test as well as differentiate the emotions of various students and thus gives the information regarding attention span of the student by analysing each image. Using various packages in python and the hadoop cluster made our project simpler, accurate and saved us a huge amount of time.

The proposed work can also be relevant to other application domains such as advertising, gaming and entertainment, where these affective states are important. Existing commercially available affective recognition systems have limited use in real-world environments, necessitating further work on this problem.

# CHAPTER - 8

# 8. FUTURE ENHANCEMENTS

The algorithm that has been developed in this project is a basic approach to what can be considered as huge component. Future enhancements for this attention span detection algorithm include capturing videos from the live session, feed and processing it in real-time using web camera of the students' laptop. Also, a Graphical User Interface can be created which enables an instructor to detect the attention of the students just by clicking on the respective name or roll number or the student instead of changing the path in the testing code.

On successfully testing the videos of the students to detect the attention span we obtained an accuracy of almost 70% which is not bad comparing the previous models. The model can be made more accurate by training more videos from the dataset into the system. This could not be achieved in our regular laptop or personal computer due to resource constraints. By integrating all these enhancements the project could be more presentable and accurate. There are a lot of features and functionalities that can be integrated in the proposed system, but the project scope has been limited to diligently resolve the problems as identified in the problem areas above. The project objective has to be achieved pertaining to the Time Constraint and Monetary constraint applied in accordance with the defined functionality of the system. We also need to improve in several areas in future to resolve the errors and improve the accuracy.

# CHAPTER - 9

# 9. BIBLIOGRAPHY

## Website references:

https://www.pyimagesearch.com/2019/07/15/video-classification-with-keras-and-deep-learning/

https://www.analyticsvidhya.com/blog/2019/09/step-by-step-deep-learning-tutorial-video-classification-python/

https://towardsdatascience.com/wtf-is-image-classification-8e78a8235acb

https://stackoverflow.com/questions/56103938/glob-got-an-unexpected-keyword-argument-recursive/56103966

Baron-Cohen, S.: Mind reading [: the interactive guide to emotions. Jessica Kingsley Publishers (2003)

Baylari, A., Montazer, G.A.: Design a personalized e-learning system based on item response theory and artificial neural network approach. Expert Systems with Applications 36(4), 8013–8021 (2009)

Brusilovsky, P: Knowledgetree: A distributed architecture for adaptive e-learning. In: Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters. pp. 104–113. ACM (2004)

Burke, A.: Crowdsourcing scientific progress: how crowdflower's hordes help harvard researchers study tb. Forbes. October 16 (2011)

Emotient: http://www.emotient.com/

Emovu: http://www.emovu.com/e/

Gunes, H., Schuller, B., Pantic, M., Cowie, R.: Emotion representation, analysis and synthesis in continuous space: A survey. In: Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on. pp. 827–834. IEEE.