ISSN No:-2456-2165

Artificial Intelligence Powered Global Learning Management System - AiTutor: A Behavioral Analytics Driven Artificial Intelligence Model Anchored In Tracking Learner-Tutor Interaction and Engagement.

Megha Vijaya Kumar[1], Ritendu Bhattacharyya[2], Adityaveer Dhillon[3], Sharat Chandra K. Manikonda[4], Bharani Kumar Depuru[5]
[1]Research Associate, Innodatatics, Hyderabad, India.
[2]Mentor, Research and Development, Innodatatics, Hyderabad, India.
[3]Team Leader, Research and Development, Innodatatics, Hyderabad, India
[4]Vice President, Innodatatics, Hyderabad, India.
[5]Director, Innodatatics, Hyderabad, India

***Corresponding Author:** Bharani Kumar Depuru
OCR ID: 0009-0003-4338-8914

Abstract:- AiTutor, the Artificial Intelligence Powered Global Learning Management System, provides an enhanced monitoring system that aims to improve the learning abilities of the students, and the teaching ability of the tutors. There has been a great increase in online platform learning from various Massive Open Online Courses (MOOCs). However, it is increasingly difficult to track the actual learning of the users in these platforms. There is always a chance that the individual who has enrolled and the individual who is learning might be different. This proxy system diminishes the efforts of people giving great effort, towards creation of content for individuals to consume. AiTutor works towards solving this problem. With its advanced Facial Recognition system, it will track whether the person who is consuming the content is the actual intended individual or is it a proxy. Along with this AiTutor would also be pivotal in tracking the attention and concentration of the learners.

The key targets for AiTutor include schools, educational organization, the tutors working for various such schools and organizations. Employers, who want to monitor the work performance of their employees, and various governmental bodies.

This research aims to give a detailed walkthrough of the product as it stands, and provide details regarding all aspects of the product. This covers the various tech stacks used, the implementation of the said technologies, the reports shown to the different end users. This provides the workflow of the product. The product demonstration will cover the working of the various AI (Artificial Intelligence) engines involved in AiTutor.

Keywords:- *Artificial Intelligence, Deep Learning, Computer Vision, Facial Recognition, Emotion Analysis, Behavioral Analysis, Learning Management System, Large Language Models.*

I. INTRODUCTION

AiTutor is a system designed to understand students' interaction with online courses taken by them. It aims at solving the issue of lack of attention span of students while taking online courses. It features various different mechanisms that aim at solving the problem of proxy attendance in online courses. There is a lack of viable solutions for the problems pertaining to proxy or malpractices when it comes to online learning in the market. This creates a perfect environment for AiTutor to thrive. A lack of viable solution creates a unique business opportunity for the product, while solving a major issue plaguing the industry. While the industry may still be a niche industry, there is an ever-increasing demand for the same.

The project methodology followed here is the open source CRISP-ML(Q) methodology from 360DigiTMG (ak.1) [Fig.1] where CRISP-ML(Q) stands for CRoss Industry Standard Practice for Machine Learning with Quality assurance. CRISP-ML(Q) can broadly be defined as a methodology designed to deal with a Machine Learning related solution's project lifecycle [13].

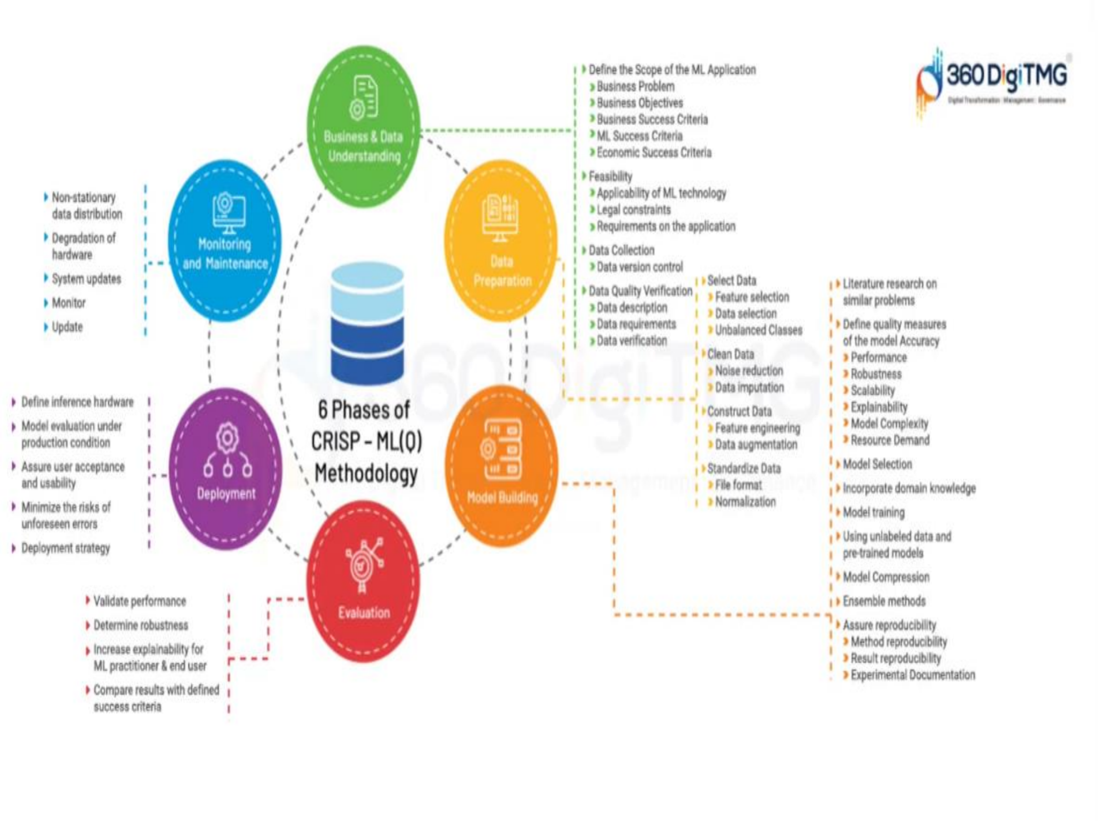


Fig.1: CRISP-ML (Q) Methodological Framework, outlining its key components and steps visually.
(Source: -Mind Map - 360DigiTMG)

Identification and understanding of the issues faced in the industry are the first task that needs to be performed under CRISP-ML(Q) [Fig.1]. AiTutor aims towards solving the problems related to attention span in an online learning class, as well as, the issue with proxy attendance in the growing sector of Online learning.

Data is the crux of any Machine learning or Deep learning solution. Since AiTutor is developed using pre-trained models, there is no requirement for primary data collection, but data from the users is collected as a onetime process to enable facial recognition. The videos collected are used for facial recognition, and once the user has been trained for access, their videos are removed to ensure that the privacy of the user is maintained. Facial Recognition has been on the rise, and its implementation has widely increased, specifically in the field of education [6]. As such, AiTutor focuses on leveraging the best of the available techniques in the field of Face Recognition technologies. AiTutor is built leveraging the facilities of ArcFace from DeepFace library [7].

Online education faces a major issue of high number of enrollments with low number of completions. In order to extract the best out of the human resource available, it is paramount to improve this ratio. One such potential solution to this problem is usage of emotion analysis for analyzing the performance, as well as, live response of the users [8, 9]. Emotion Analysis stands out as a great tool of understanding whether a person is able to grasp the concepts being taught [9].

The time saved due to the usage of pre-trained models was instead leveraged towards the deployment and monitoring & maintenance phase of CRISP-ML(Q) [Fig.1]. The deployment was done by leveraging the two tools, Amazon Web Services (AWS) Elastic Cloud Compute (EC2), and Apache2 web server [14]. The users will be able to see the final reports as part of their frontend UI, while the data generated from AiTutor will be stored in AiTutor.

II. METHODS AND TECHNIQUES

A. Data Dimension

Below are the details which were used for facial recognition.

Number of Videos	43
Number of images	2150
Size of video data	500 MB

Apart from facial recognition we took the leverage of pretrained models. So directly it was implemented in the API.

B. Model Architecture

In this research, we elucidate the intricate details of our project's architecture and its pivotal role in advancing our research endeavors.

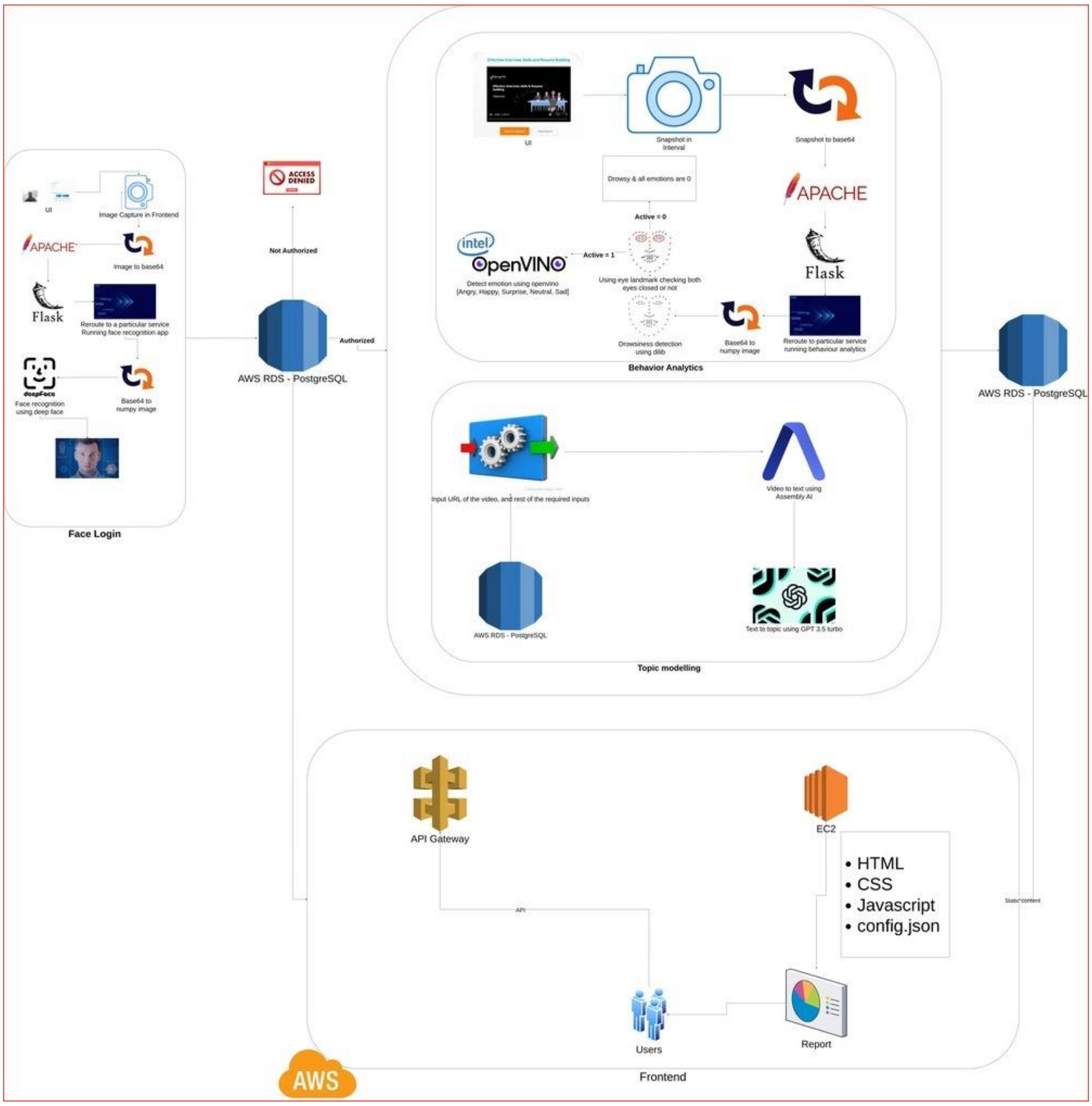

UI
Image Capture in Frontend
APACHE
Image to base64
Flask
Reroute to a particular service Running face recognition app
deepFace
Face recognition using deep face
Base64 to numpy image
Face Login
ACCESS DENIED
Not Authorized
AWS RDS - PostgreSQL
Authorized
UI
Snapshot in Interval
Snapshot to base64
Drowsy & all emotions are 0
Active = 0
APACHE
intel OpenVINO
Active = 1
Detect emotion using openvino [Angry, Happy, Surprise, Neutral, Sad]
Using eye landmark checking both eyes closed or not
Flask
Drowsiness detection using dlib
Base64 to numpy image
Reroute to particular service running behaviour analytics
Behavior Analytics
Input URL of the video, and rest of the required inputs
Video to text using Assembly AI
AWS RDS - PostgreSQL
Text to topic using GPT 3.5 turbo
Topic modelling
AWS RDS - PostgreSQL
API Gateway
EC2
• HTML
• CSS
• Javascript
• config.json
API
Static content
Users
Report
AWS
Frontend

Fig. 2: Architecture Diagram: Explain the workflow of the behavior analytics module of AiTutor
(Source :ML Workflow - 360DigiTMG)

The deployment of AiTutor is flexible and designed to make it easy for the modules to be scaled as per need. The Artificial Intelligence modules are separated into individual modules, so as to ensure that all the modules are independent of each other.

The modules are deployed on an AWS EC2 instance, with the individual modules being deployed using either flask or FastAPI [14] [Fig.2, 3]. The Flask and FastAPI Application Programming Interfaces (APIs) are designed to run as a local application on various different ports within the same EC2 instance. The modules are separated using the virtual environment package, venv, from python.

To ensure that the applications are running at all times, without running the explicit python commands, they are run via Linux service. Service in Linux is used to start, stop or restart a daemon or script. In AiTutor's case it is used to run the scripts. This will ensure that the modules are running at all times required in the background. Apache2 web server acts as the middleware to communicate with the frontend, and the Linux services running the AiTutor modules [Fig.2].

The frontend will capture the information of the user, in the form of a snapshot taken by the frontend. Leveraging JavaScript, the image is converted into base64 format. The base64 converted image is then received by the Apache web server. Apache by default will run an application on port 5000. This is a flask application, which will receive all the traffic from the frontend, and it will also be sending out the output generated by the AI modules in the backend [Fig.2].

Once the base64 data is received by the Apache application, it will look at the Uniform Resource Locator (URL) of the frontend. Using the URL, the application is written in such a way that it can recognize which AI module should be receiving the data. The data will be forwarded to the relevant backend module. Since, AiTutor is dealing with image related data from the frontend, it will follow the same preprocessing steps once the data is received by the backend modules. The images will get converted from base64 format, back to aNumPy array. NumPy is a python module used to work with arrays [Fig.2, 3].

In Facial Recognition, the image is received, and if the user is recognized the access is provided to the user. In Behavior Analytics, it will check for active/drowsy status, and the emotions displayed by the user to generate and send the output back to a Postgres database. The database will store this information, and if the user sends a request for the report, the information is gathered by the behavior analytics service in Linux, and sent to the user via the Apache web server pipeline [Fig.2, 3].

Topic model generation part is a separate application. That application takes the URL, start time, end time, and temperature, on the basis of these inputs, it extracts the text from the video using Assembly AI. Using GPT models it generates the topics on the basis of the extracted texts. It saves in the database, which gets populated in the AiTutor application.

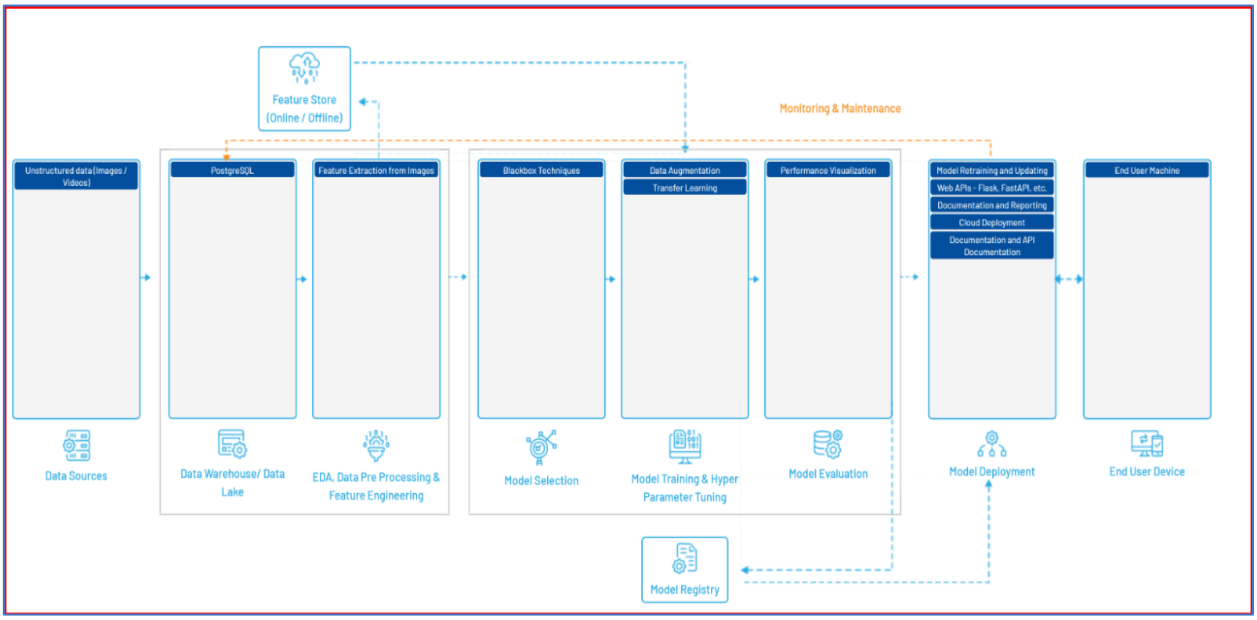

Feature Store
(Online / Offline)
Monitoring & Maintenance
Unstructured data (Images / Videos)
PostgreSQL
Feature Extraction from Images
Blackbox Techniques
Data Augmentation
Transfer Learning
Performance Visualization
Model Retraining and Updating
Web APIs - Flask, FastAPI, etc.
Documentation and Reporting
Cloud Deployment
Documentation and API Documentation
End User Machine
Data Sources
Data Warehouse/ Data Lake
EDA, Data Pre Processing & Feature Engineering
Model Selection
Model Training & Hyper Parameter Tuning
Model Evaluation
Model Deployment
End User Device
Model Registry

Fig. 3: This Machine Learning Architecture diagram illustrates the stages involved in AiTutor. (Source: - Open-Source ML Workflow Tool- 360DigiTMG)

C. Implementation of Artificial Intelligence solutions in AiTutor

AiTutor is an innovative solution, which is used for analysis of the participant's behavior through video analytics. It then generates comprehensive reports for learners as well as tutors on behaviors exhibited during the learning process. AiTutor will provide a report for the different behaviors displayed by the user. Just the presence of a person is not sufficient to understand their level of learning. It is also important to ensure that they are actually listening to what is going on in the class. One such tool is their attention span, which is captured by the Active-Drowsy model. The Attention model is tracked when the learner is active in the class, and when they are displaying drowsiness [10]. The learners have the option to watch the portion of the video during which they were sleepy.

The emotions are captured using the emotion model, and it captures five emotions which are happiness, sadness, angriness, surprise and neutral emotion. Neutral is the base emotion in case the learner is giving indistinguishable emotions [8, 9]. The development of a comprehensive emotion model is difficult, as such pretrained models are used to save time and cost of implementation. The models chosen for this are provided by Intel's OpenVino toolkit (ak.2).

The tutors are given the reports related to the engagement, or lack thereof, of the learners. The tutor can then use these reports to enhance the teaching methods implemented.

The product is deployed on Amazon Web Services (AWS), and it allows greater flexibility in terms of deployment of the product on various platforms. It also ensures that each individual group of users is given an isolated system, to avoid any system conflicts. One of the most critical aspects of the deployment is the fact that the model has the capability of running on both Central Processing Unit (CPU) as well Graphical Processing Unit (GPU). This allows for cost saving measures to be put in place depending on user requirements[Fig.2].

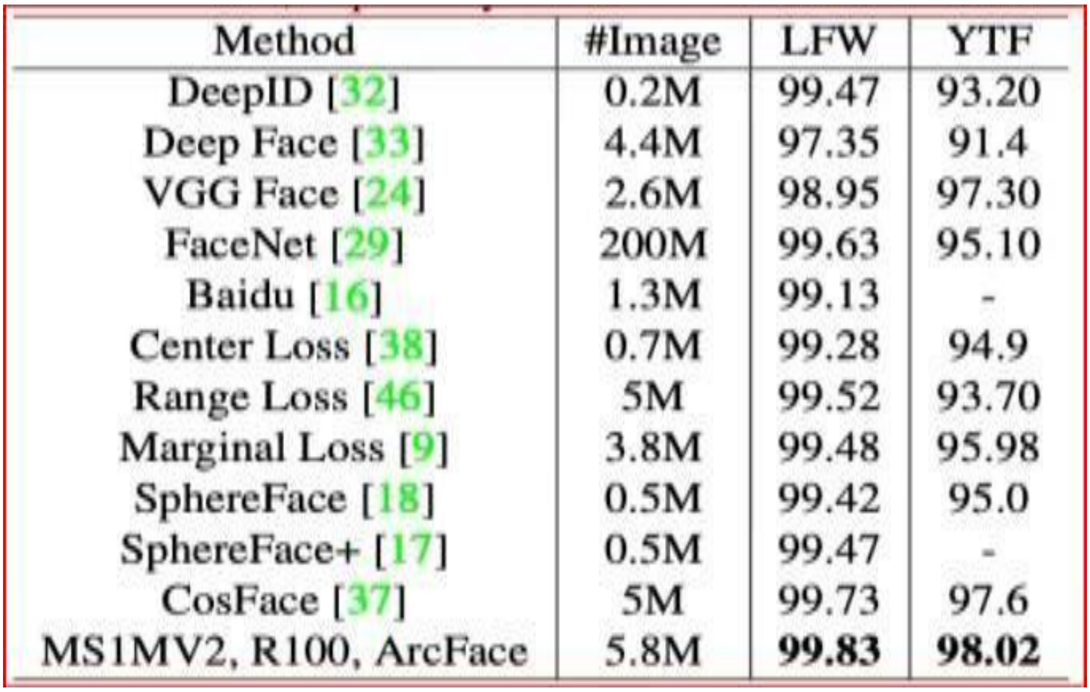
AiTutor will be able to monitor performance of whoever is in front of the system. This will allow the mechanism to not only perform analysis on the user, and provide detailed working reports for the same, it will also enable the system to keep a track of whether the person using the online learning system is the registered user. This reduces the risk of malpractice for the online platforms,

The product will impact various different users from individual learners, individual tutors, to Multi-National Corporations (MNCs), and governmental organizations. This will allow for a constant feedback loop in the mechanism, which will allow for timely and regular updates in the courses offered.

This allows organizations to optimize their upskilling programs for their employees, and it will also improve the overall learning experience of the individual users. Having such an optimization will lead to overall better performance from the individuals across the board, thus leading to economic benefits, such increase in ROI, and reduction of time taken to perform complex tasks.

The entire AI engines are developed using open-source platforms. This reduces the development cost of the product and in turn reduces the overall production cost of the system.

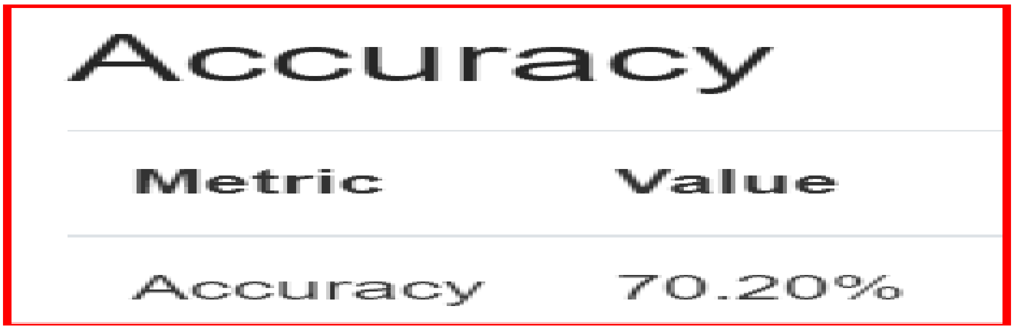
Facial recognition software is developed using ArcFace from Deep Face library [2, 3, 7]. All the users are given access to the platform via their trained faces. It uses a similarity matrix that uses Angular Margin Loss instead of SoftMax loss. ArcFace achieves an accuracy of 99.83% in comparison to some of the other open source models[Fig.4].



Method	#Image	LFW	YTF
DeepID [32]	0.2M	99.47	93.20
Deep Face [33]	4.4M	97.35	91.4
VGG Face [24]	2.6M	98.95	97.30
FaceNet [29]	200M	99.63	95.10
Baidu [16]	1.3M	99.13	-
Center Loss [38]	0.7M	99.28	94.9
Range Loss [46]	5M	99.52	93.70
Marginal Loss [9]	3.8M	99.48	95.98
SphereFace [18]	0.5M	99.42	95.0
SphereFace+ [17]	0.5M	99.47	-
CosFace [37]	5M	99.73	97.6
MS1MV2, R100, ArcFace	5.8M	**99.83**	**98.02**

Fig. 4: The above table illustrates the accuracy of ArcFace in comparison to other open-source facial recognition models.

Emotion analysis model [4, 5] is open source and is provided by Intel (ak.2). It is a Convolution Neural Network (CNN) model that captures five emotions i.e., neutral, happy, sad, surprise and angry. The model achieves an accuracy of 70% in terms of emotions detection [Fig.5].



Accuracy

Metric	Value
Accuracy	70.20%

Fig. 5: The above table is showing the accuracy of the emotion analysis model used

Dlib Facial landmarks are used to extract the information for drowsy recognition. Dlib collects 68 landmarks and from that the landmarks for left and right eye are gathered. The Eye Aspect Ratio (EAR) is calculated to detect whether a person is active or drowsy. EAR is a simple, yet robust technique used to check the blinking of the eye, as well as, the level to which the eye is opened[11, 12] [Fig.6].

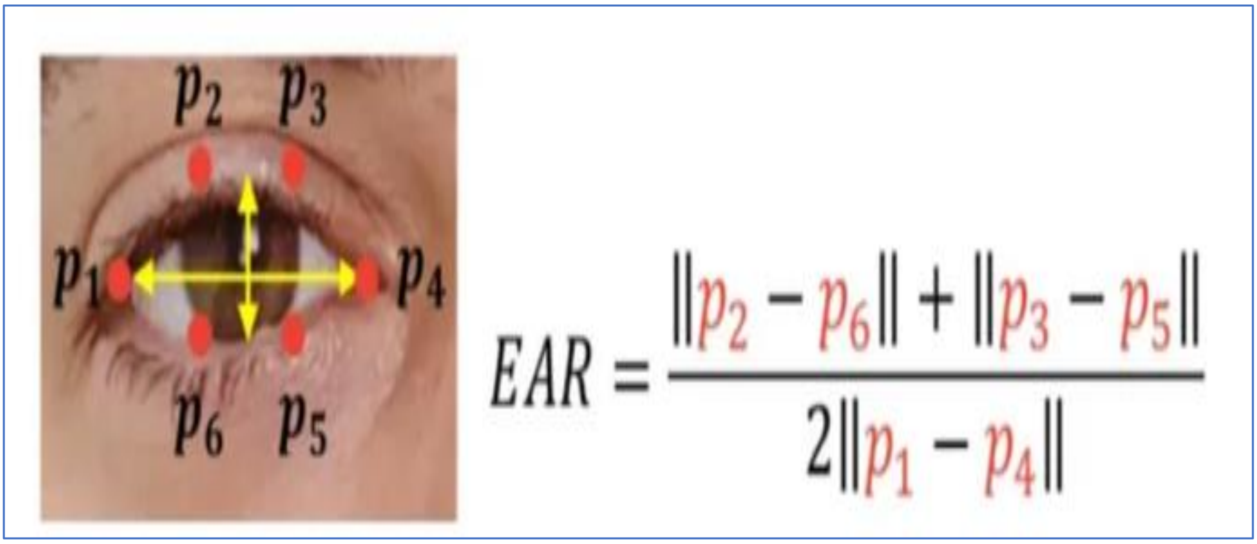

p_2
p_3
p_1
p_4
p_6
p_5
$EAR = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\|p_1 - p_4\|}$

Fig.6: The image here is showing the eye landmarks captured by Dlib and the Eye Aspect Ratio mathematical formula used. (Source - Eye Aspect Ratio (EAR) and Drowsiness detector using dlib)

EAR is calculated taking the distance between the vertical points as well as horizontal points of the eyes. Once it is calculated, the model will make a judgment based on the set threshold and give the output as active or drowsy

- **Frontend:** Frontend is a combination of JavaScript (JS), Cascading Style Sheets (CSS) and Hypertext Preprocessor (PHP). The frontend shares the images from the User Interface (UI) and receives the processed information for analysis [1].
- **Middleware:** AWS Elastic Cloud Compute (EC2) instance, acts as the overall host for the system and then Apache2 web server allows the application to communicate back and forth between the frontend and backend. It allows the data to move between the UI and backend, allows for model processing and then sharing of the output [1].
- **Backend:** in the backend we have python codes, which contain all the models and weights files, as well the local deployment codes using Flask and FastAPI framework [1].
- **Database:** The database used is Postgres. All the details regarding the database are contained on AWS Relational Database Services (RDS) service. The server contains all the prerequisite information like details regarding the topics being played in the video, and it will also store the processed information as when it is generated.

Like all AI platforms there are challenges faced while developing the system. One key problem faced early on was the distinct lack of Indian faces in various open-source face recognition models. This can be overcome by using a greater number of images while training the face recognition system, as well as, using data augmentation techniques and backend, allows for model processing and then sharing of the output [1].

III. RESULTS AND DISCUSSION

The aim of the AI system is to provide detailed reports to display the results for the user. Enhanced reports will allow for better understanding of the analysis performed by the AI engine. This will improve the overall experience of the user in relation to AiTutor

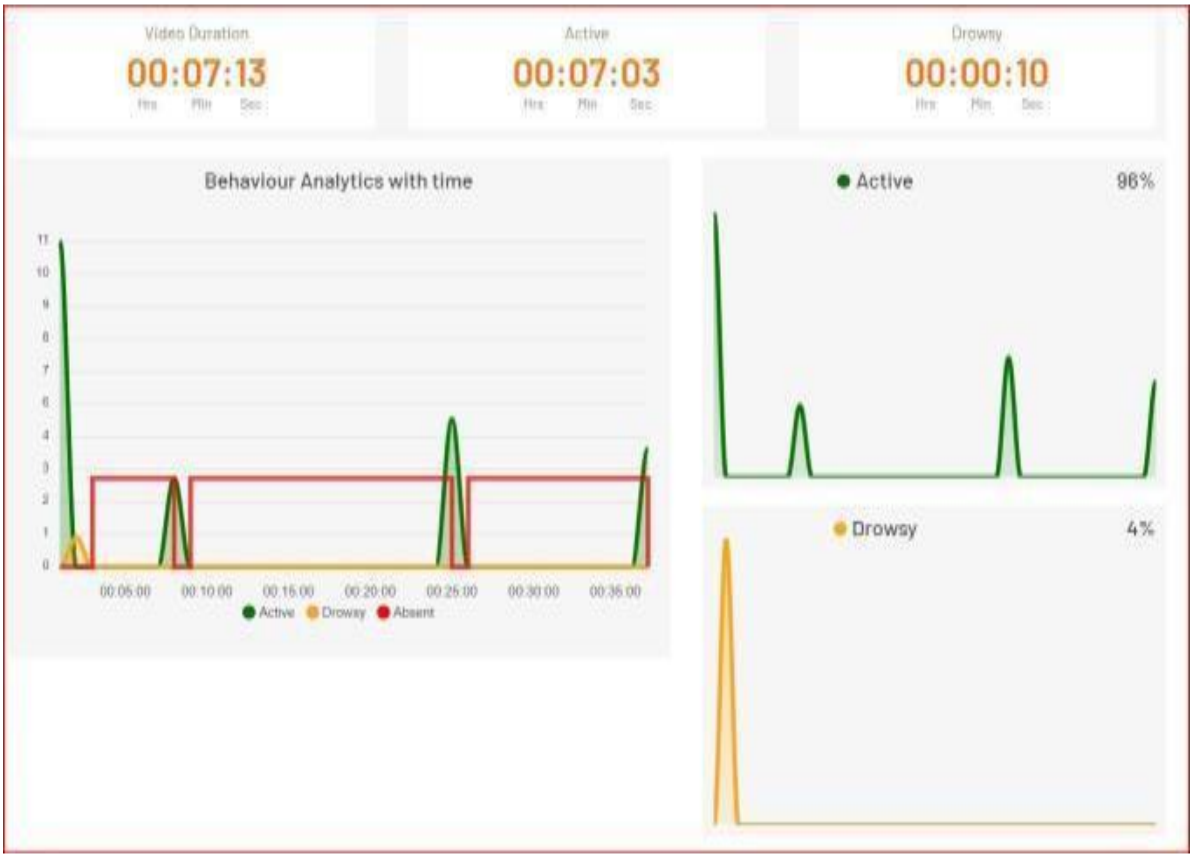


Fig. 7: The above report shows the active drowsy status of the user. It also displays the time the user was absent from the video. It will show the total time it takes the user is active/drowsy and it also gives the output as the total time in percent.

In [Fig.7], the user will be able to understand how long they were watching the video, and for how long they were active in that video. They use this knowledge to understand the efficiency of their learning experience.

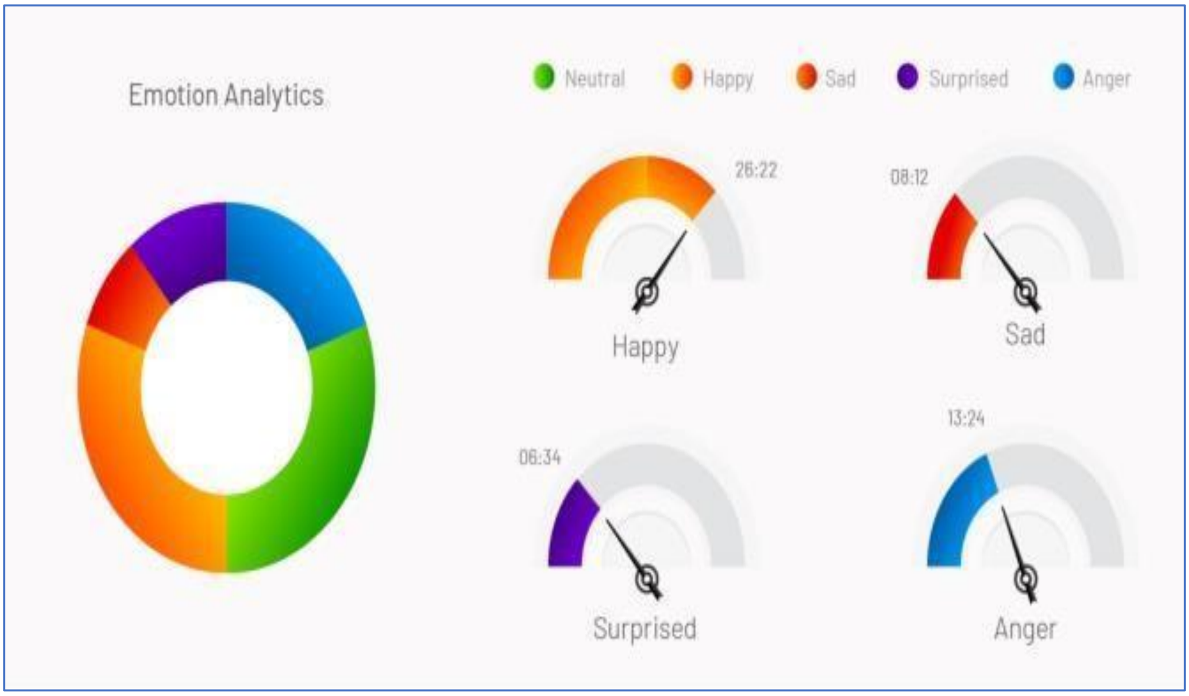
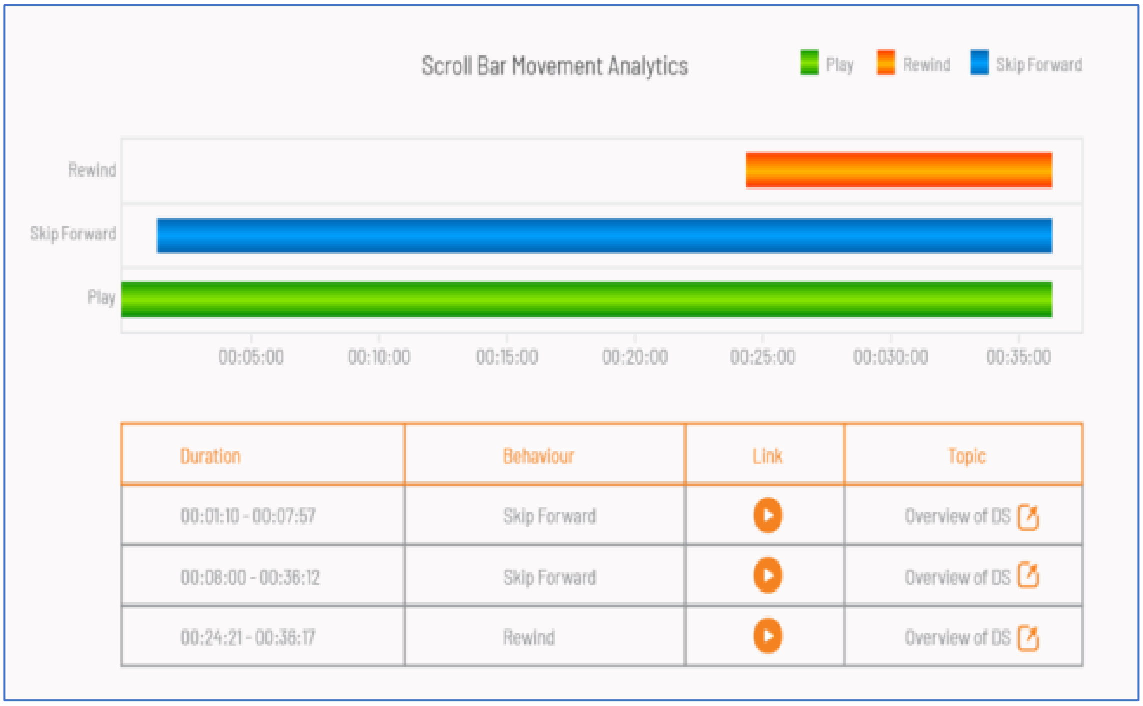


Fig. 8: The above pie chart shows the overall breakdown of the emotions displayed by the user. This report is further enhanced by the speed-o-meter of the various emotions. This will show the total time per emotion during a session, and then percent value of the same

The user will get a pie chart and speed-o-meter [Fig.8], to grasp a better understanding of their emotions. Emotions play a major role in education, just as they do in all sectors of human society. Having a stable emotions base can lead to enhanced learning. While this report is a small part of understanding the overall emotions of the user, it can guide the user on the right path. By analyzing their emotions, they can understand what sort of positive and/or negative effects their emotions might have on the learning process.



Duration	Behaviour	Link	Topic
00:01:10 - 00:07:57	Skip Forward	▶	Overview of OS
00:08:00 - 00:36:12	Skip Forward	▶	Overview of OS
00:24:21 - 00:36:17	Rewind	▶	Overview of OS

Fig. 9: The scroll bar analysis of the user. If the user is skipping forwards or backwards, this will capture that. The videos are stored in a database, with proper segmentation of the topics. So, the user can click on the link and go to the point from where they skipped the video or rewind it.

The user also gets an analysis on how they interact with the scroll bar while watching the video [Fig.9]. They will be able to get a comprehensive view of when they were skipping forward or rewinding the video. They will also be able to see the topics they skipped forward or backward. These topics are generated using topic modeling.

In Topic modeling workflow, the process begins with the consideration of an input video URL. The corresponding .m3u8 file is retrieved through script operations. This .m3u8 file, encoded in UTF-8, operates as a playlist, containing vital information such as URL paths for streaming videos and media details. Extracting the audio from the video involves utilizing the video path obtained from the .m3u8 file. The tool considered for this task is FFmpeg, an open-source software known for its audio extraction capabilities. FFmpeg adeptly disables the video component from the input, focusing on processing the audio into a 16-bit PCM (Pulse Code Modulation) encoding, producing a standalone audio file.

With the audio file, the next objective is audio-to-text conversion. Assembly AI, a tool proficient in transcription, becomes pivotal for this purpose. Leveraging Assembly AI's API, the audio file undergoes transcription, requiring appropriate authorization and API keys. The API processes the audio file, incorporating transcription start and end times in milliseconds. The response is returned in JSON format, encapsulating crucial details like language model, language code, acoustic model, text, audio duration, and confidence level. The 'text' key holds the complete transcription, while the 'words' key furnishes timestamps and confidence scores for each word uttered in the audio, offering a comprehensive transcription dataset.

This transcribed text, a valuable resource, becomes a foundation for subsequent analysis, specifically, topic modeling. Employing OpenAI's GPT (Generative Pre-trained Transformer) model, topics of discussion are generated from the transcribed text. OpenAI's chat completion API, catering to both GPT-4 and GPT-3.5-turbo models, plays a central role in this task. The transcribed text is initially considered as the context, and a relevant prompt is then passed to the model, triggering the generation of topics based on the provided text. This collaborative integration of audio processing, transcription, and AI-driven topic modeling forms an effective and cohesive workflow for comprehensive analysis and understanding of the content within the given video.

Using these reports individually has many benefits, but the real benefit will arise when the reports will be used in conjunction with each other. As these reports complement each other, getting an overall report and analyzing the same information, will help the users as well as the tutors to improve their learning experience.

IV. CONCLUSION

The implementation of AiTutor looks towards filling a major gap in a growing sector of the industry. As more and more of the world digitized, the scope of online courses will continue to grow. AiTutor unveils a multitude of benefits, ranging from personalized learning experiences to dealing with any potential issues that a student might face while undergoing their learning. There are always going to be concerns regarding the ethics of such a mechanism, that's why AiTutor ensures that the users personal information is not stored for any reason whatsoever.

The fusion of Artificial Intelligence with human intelligence has the potential to revolutionize an entire industry, fostering a more emphatic and responsive learning environment for everyone involved. In this pursuit, AiTutor represents a crucial stepping stone towards a brighter future, more emotionally aware future

ACKNOWLEDGMENTS

- We acknowledge that with the consent from 360 Digi TMG, we have used the CRISP-ML(Q)and the ML Workflow which are available as open-source in the official website of 360DigiTMG methodology (ak.1)
- We acknowledge the usage of Intel OpenVino Toolkit Available for download (ak.2)

REFERENCES

[1]. Gibran Benitez-Garcia, Tomoaki Nakamura and Masahide Kaneko Journal: Journal of Signal and Information Processing, 2017, Volume 08, Number 03, Page 132 DOI: 10.4236/jsip.2017.83009.

[2]. Facial expression recognition using deep convolutional neural networks Published in: 2017 9th International Conference on Knowledge and Systems Engineering (KSE) INSPEC Accession Number:17393763,DOI: 10.1109/KSE.2017.8119447.

[3]. Richa Grover, Sandhya Bansal, "Facial Expression Recognition: Deep Survey, Progression and Future Perspective", 2023 International Conference on Advancement in Computation & Computer Technologies (InCACCT), pp.111-117, 2023.

[4]. ViplavPandhurnekar, Anish Iyyappan, DnyandeepDhok, Vaishnav Khante, SampadaWazalwar, "Proposed Method for Threat Detection Using User Behavior Analysis", 2023 IEEE 3rd International Conference on Technology, Engineering, Management for Societal impact using Marketing,Entrepreneurship and Talent (TEMSMET), pp.1-5, 2023.

[5]. Rengarajan R, Shekar Babu, "Anomaly Detection using User Entity Behavior Analytics and Data Visualization", 2021 8th International Conference on Computing for Sustainable Global Development (INDIACom), pp.842-847, 2021.

[6]. Mark Andrejevic& Neil Selwyn (2020) Facial recognition technology in schools: critical questions and concerns, Learning, Media and Technology, 45:2, 115-128, DOI: 10.1080/17439884.2020.1686014.

[7]. J. Deng, J. Guo, J. Yang, N. Xue, I. Kotsia and S. Zafeiriou, "ArcFace: Additive Angular Margin Loss for Deep Face Recognition," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 44, no. 10, pp. 5962-5979, 1 Oct. 2022, doi: 10.1109/TPAMI.2021.3087709.

[8]. Wang L, Hu G, Zhou T. Semantic Analysis of Learners’ Emotional Tendencies on Online MOOC Education. Sustainability. 2018; 10(6):1921. https://doi.org/10.3390/su10061921.

[9]. Weiqing Wang, Kunliang Xu, HongliNiu, Xiangrong Miao, "Emotion Recognition of Students Based on Facial Expressions in Online Education Based on the Perspective of Computer Simulation", Complexity, vol. 2020, Article ID 4065207, 9 pages, 2020. https://doi.org/10.1155/2020/4065207.

[10]. Poojari, N.N., Sangeetha, J., Shreenivasa, G., Prajwal (2022). Automatic Student Attendance and Activeness Monitoring System. In: Reddy, V.S., Prasad, V.K., Mallikarjuna Rao, D.N., Satapathy, S.C. (eds) Intelligent Systems and Sustainable Computing. Smart Innovation, Systems and Technologies, vol 289. Springer, Singapore. https://doi.org/10.1007/978-981-19-0011-2_36

[11]. S. Sathasivam, A. K. Mahamad, S. Saon, A. Sidek, M. M. Som and H. A. Ameen, "Drowsiness Detection System using Eye Aspect Ratio Technique," 2020 IEEE Student Conference on Research and Development (SCOReD), Batu Pahat, Malaysia, 2020, pp. 448-452, doi: 10.1109/SCOReD50371.2020.9251035.

[12]. CaioBezerraSoutoMaior, Márcio José das Chagas Moura, João Mateus Marques Santana, Isis Didier Lins, Real-time classification for autonomous drowsiness detection using eye aspect ratio. Expert Systems with Applications, Volume 158, 2020, 113505, ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2020.113505.

[13]. Studer, S.; Bui, T.B.; Drescher, C.; Hanuschkin, A.; Winkler, L.; Peters, S.; Müller, K.-R. Towards CRISP-ML(Q): A Machine Learning Process Model with Quality Assurance Methodology. Mach. Learn. Knowl. Extr. 2021, 3, 392-413. https://doi.org/10.3390/make3020020

[14]. Ahmed, Khandakar& Islam, Motaharul. (2022). A Comparative Analysis of AWS Cloud-Native Application Deployment Model. https://doi.org/10.1007/978-981-19-2445-3_29.