Traitement du Signal
Vol. 42, No. 2, April, 2025, pp. 943-961
Journal homepage: http://iieta.org/journals/ts

Detecting Student Engagement with Convolution Neural Network and Facial Expression Recognition

Nuha Alruwais, Mohammed Zakariah*

Department of Computer Science and Engineering, College of Applied Studies and Community Services, King Saud University, Riyadh 11495, Saudi Arabia

Corresponding Author Email: mzakariah@ksu.edu.sa

https://doi.org/10.18280/ts.420229

Received: 25 April 2024
Revised: 4 November 2024
Accepted: 16 November 2024
Available online: 30 April 2025

Keywords:
student engagement, deep-learning, engagement recognition, Facial Expression Recognition (FER), online-learning, student performance

ABSTRACT

Student engagement is a fundamental idea in modern education viewed as a goal. This study investigates ways to detect students' engagement automatically based on facial expressions. The Facial Expression Recognition (FER) dataset is utilized in this study to assess how healthy ML models identify facial expressions in images. It contains around 35,000 to 40,000 images and has become a typical benchmark for evaluating the efficiency of Machine Learning (ML) algorithms. The images in the collection are primarily grayscale and have been scaled to 48×48 pixels. The images were gathered from a varied community of various races, genders, and ages. It is possible to assess how well the CNN model was trained to distinguish between engaged and disengaged students in the context of students' attention detection using the FER dataset. The AUC-ROC curve is a scalar form of the ROC curve that demonstrates the classifier's overall ability to differentiate between negative and positive classifications. Moreover, our model's ROC is 1, indicating flawless categorization of both visible and unseen data, with all other variables operating uniformly. According to the results, our CNN model obtained 94% accuracy, which is higher than other models due to its innovative architecture, enhanced optimization technique, and extensive dataset. Other papers and model accuracy indicate that our model outperforms the other models listed in the research. While performing similarly to our model, the previous CNN model, which uses the FER dataset and a CNN architecture, nonetheless received an 89% accuracy. In comparison, models with 57% and 72% accuracy are substantially lower. The complexity of the model, hyperparameter tuning, and data calibration are all factors that may impact model performance.

1. INTRODUCTION

For e-learning systems, predicting students' engagement in a class is essential [1]. The tool enables the instructor to observe how the student interacts with numerous aspects. The learning outcomes will increase with the depth and duration of the student's engagement in the lesson. Research in educational theory has always focused heavily on student engagement [2]. Early attention to engagement was prompted by concerns about high dropout rates and data showing that many students, between 30% and 70%, reported feeling bored and disengaged in the classroom regularly. The amount of student interest or disengagement might be determined using eye movements [3], facial expressions [4], gaze patterns, eye tracking, and body movements [5]. An outside observer can evaluate how engaged people are.

Analyzing student engagement could help accurately measure the mentioned characteristics. The advancement of student engagement is helpful for online learning and teaching in conventional classrooms, innovative and challenging educational games, and online tutoring platforms [6, 7]. Any learning environment may be considered a combination of cognitive, behavioral, and emotional states regarding student engagement. Active engagement and participation of the student in class activities are necessary for behavioral engagement to occur. On the contrary, behavioral engagement involves students' grasping information and self-dedication to the learning process.

In contrast, emotional engagement indicates the student's affective state and direct engagement in the class [8-10]. Emotionally and intellectually engaged students are more likely to work more, persevere longer, and meet course requirements more successfully than less involved students. As a result, this component is essential in defining each student's welfare and growth. Emotional engagement in interactions with instructors and peers indirectly modifies children's increased cognitive involvement [10]. The level of student engagement affects upper-body posture, facial expressions, and general ecological factors. However, in an online learning environment, facial expressions are the most intuitive behavioral way to demonstrate interest [11]. For the most part, datasets included the seven primary emotions of grief, pleasure, neutrality, surprise, disgust, wrath, and contempt [11-13]. Now, the emphasis is on monitoring students' emotions during learning to gather particular information about their engagement. Recent research has linked facial expressions to self-reported and assessed emotional conditions related to learning, including boredom, engagement, confusion, and frustration [12]. In a synchronous learning scenario, it is important to consider the connection between the intensity and timing of facial expressions [13].

ML and Deep Learning (DL) are used in two approaches to measure affective expression. Using hand-drawn patterns, the ML-based technique gathers facial data to determine engagement levels [14]. In emotional state prediction tests, DL techniques outperform traditional ML-based methods because they learn features from training data, permitting the algorithm to recognize perfectly all right facial differences [14, 15]. Additionally, DL-based systems are non-intrusive. The tools they utilize to collect and analyze facial expression video data are accessible, controlled, and straightforward. The two types of DL techniques are now used for emotion identification: Video sequence-based techniques and approaches based on static images [15]. Classifying facial emotions from a sequence of linked frames in a video is more natural because a video sequence provides FER with substantially more information than a static image of a person's face. There are three different types of video sequence-based approaches: 3D CNN, which is a combination of CNN and Long Short-Term Memory networks, as well as a combination of spatial and temporal CNN models [16, 17].

Researchers created a DL model comprising two critical steps to overcome the abovementioned issues: fundamental facial expression identification and student engagement recognition. The FER-2013 dataset was initially used to train a CNN for a facial representation model and achieve good performance. A student engagement recognition model was created using a separate CNN and trained on a compiled student engagement recognition domain dataset. Then, we employed the model to activate it. Figure 1 illustrates the conceptual framework for student involvement in an e-learning system that uses the DL technique. It begins by sending a dataset named FER2013 to the place where the data is being collected for further analysis or processing. On the other hand, students' engagement in learning activities starts with online learning. The data is then pre-processed before being passed to a deep-learning classifier like CNN or LSTM. Pupils are either engaged or disengaged when they accept a learning approach.

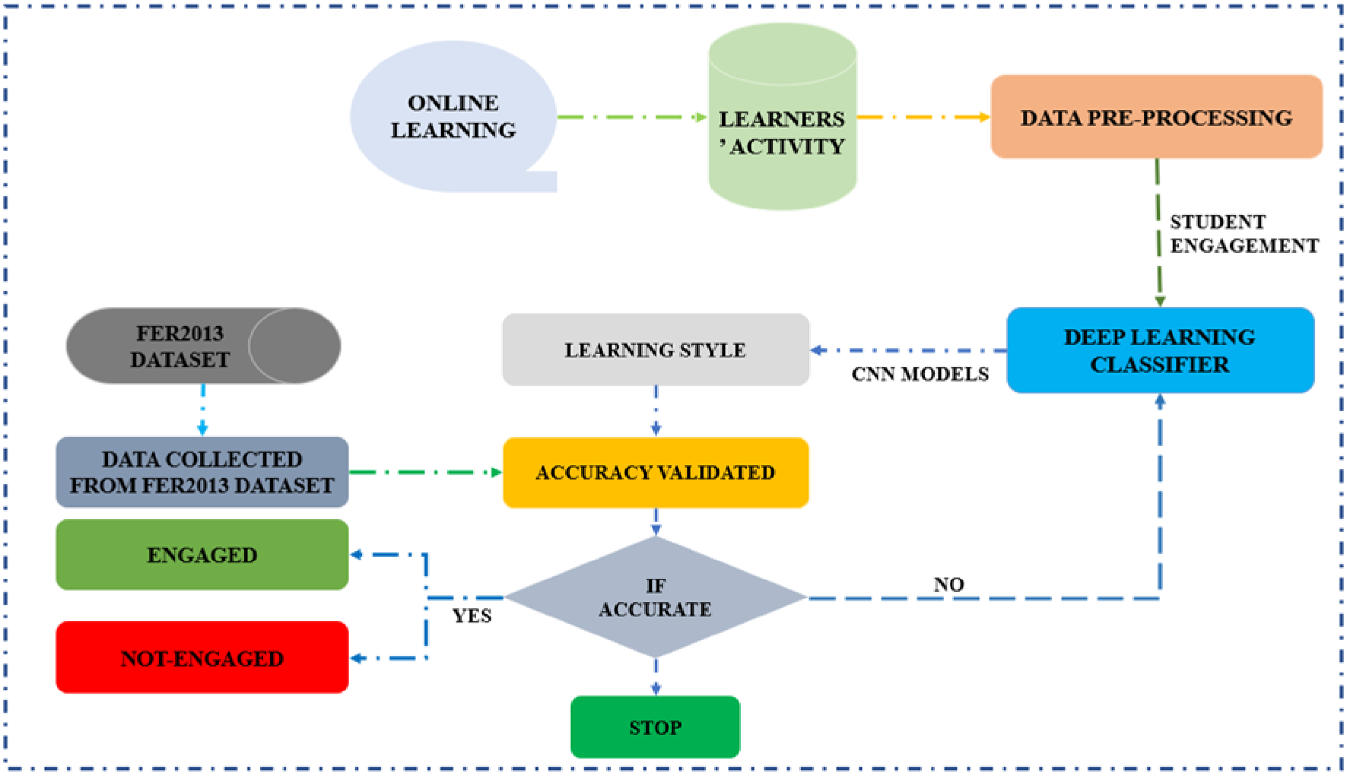


Figure 1. Deep learning as a conceptual framework for student engagement in an e-learning system

As a remedy, we developed a DL-based approach that offers our representation model, particularly for identifying student engagement. Secondly, to compensate for the lack of engagement data, we pre-train our engagement recognition model using the FER-2013 dataset, which is bigger than the dataset we gathered. This study makes several contributions:

This study described a more complicated CNN structure with extra layers that is better at reading students' facial expressions to figure out how engaged they are than the most recent top-of-the-line models.

The face representation model used in this work was successfully put into action to start up a contact recognition model and record basic facial expressions. This shows that basic facial expression data can be used to find out if a kid is interested.

This study uses a bigger, more varied dataset with repeated comments to make it easier to study how to use facial expressions to recognize students' interest in images. The dataset clears up the idea's many complexities and ambiguities by separating the emotional and behavioral parts of involvement.

New deep-learning methods are used in this study. In the FER dataset, the proposed model does better than many standard methods.

The remaining sections of the paper are as follows: The pertinent studies for identifying student engagement are described in Section 2. The experiment's datasets are covered in Section 3. The methodology and proposed framework are covered in Section 4. The experimental results following the use of our recommended approach are shown in Section 5. Student engagement is covered in Section 6. In Section 7, we finally look at the conclusion.

2. LITERATURE REVIEW

Numerous studies have evaluated how well students participate in both traditional and online learning settings. The link between student data and student involvement has been assessed in this study using a variety of approaches and input elements. A DT algorithm was used by Lan and Hew [10] to identify student engagement through non-verbal behavior. Ten students contributed the data, divided into 65 cases with 13 yes responses, 15 neutral responses, and 25 no responses. They employed five characteristics: the shoulders, face, lips, interest, and eyes. They categorized the responses as disinterested, interested, or neutral. With the help of the random DT, the maximum accuracy was 86%. A CNN model was created by Saurav et al. [11] to identify sleepy drivers. They added their collection of 4900 photos to the ZJU dataset of "open" and "closed" eye photographs (2565 open-eye images and 2568 closed-eye images). Their CNN model has an accuracy rate of 93%. To localize student involvement, Huang et al. [12] created an LSTM and DNN. Around 200 movies were taken, and 80 individuals (51 men and 29 women) provided the data. Each video lasted about four minutes. Using OpenFace, characteristics related to eye look and head position were retrieved. They employed RF, DNN, and LSTM for training. The Mean Square Error (MSE) for LSTM and RF was 0.08%.

DL models occasionally recognize facial emotions in images [18-20]. The sophisticated, multi-layered topologies of neural networks allow them to learn classified structures from low-level to high-level feature representations. To recognize facial emotions, Pan et al. [21] used CNNs and won the 2013 ER challenge. Zhang et al. [15] trained a second CNN model to recognize facial expressions, followed by a linear SVM. This model took first place in the 2013 FER competition. A multi-modal data structure was employed [17] to extract visual and auditory elements. To create captions for images with human faces that are more indicative of real people, Rajabalee et al. [16] employed a CNN model to identify facial expressions. The representation learned was utilized in the image captioning model.

In another study [21], the authors examined student engagement by comparing student effort and academic accomplishment. In this study, they gave advisers, teachers, and students access to automated feedback tools. To assess how well the Virtual Learning Environment (VLE) interacted with students, another research employed demographic information from OULAD [14]. The k nearest neighbor (KNN) and Naive Bayes (NB) networks outperform the CART and NB networks with the smaller dataset used to make the predictions. A DT was used to create a model that predicts student success using university data by extracting information in a time-series fashion [22, 23]. Early dropouts and students who require extra care might be identified with this aid [24, 25]. According to research comparing four decision tree algorithms, the J48 method is superior to other algorithms in accurately identifying poor pupils [26].

Clustering has been employed for predictive analytics, although most work focuses on classification models. A data stream classification approach called incremental semi-supervised fuzzy clustering is employed in a case study of the educational data mining process [25]. To extract a classification model that can forecast the students' results, the OULAD has been processed as a data stream. The categorization model changes due to the suggested clustering algorithm's ability to adjust to newly arriving data. Promising findings emerged from the initial numerical research. There is evidence in the literature that analysis has been done on data from sources other than e-learning logs, such as social media [27]. The student traits under study differ due to the varying sources. Data from 7.2 million video sessions across four courses on the edX platform was used in the largest-scale study yet on students' engagement with videos. Student engagement is assessed by looking at how students watch long films and how they respond to the post-video quiz. Variables like student gender and age are used to evaluate student engagement. They must be controlled because they are dependent [28].

In a similar study [29], the authors suggested a technique to identify student engagement, employing two models and eye movement and emotion analysis. The first CNN model was trained using eye images to distinguish between "focused" and "distracted" eye pictures. Grayscale images of facial emotions like "angry" and "happy" make up the second pre-trained model, which was trained using the FER-2013 dataset. In the study [30], the authors developed a CNN model to recognize student engagement and facial expressions. The engagement recognition (ER) dataset, which they created, had 4700 annotated pictures, 2389 of which were engaged and 2400 were not. The FER-2013 dataset was used to analyze facial expressions. Observing student behaviors like glancing at the keyboard or the screen was possible. They simultaneously recognized the student's facial expression, such as happiness or excitement. Then, they compared the two models to determine whether or not the student was engaged. The detection was said to be deactivated when the student was not staring at the screen or when he closed his eyes. The emotion was then labeled as bored or perplexed. Their datasets were used to train the VGG and CNN models. The engagement model has 75.68% categorization accuracy. A real-time approach to identifying the direction of an eye's gaze in films was suggested by Wang et al. [31]. Thirty-six children, 27 of whom did not have an autism diagnosis and 11 of whom did—and 45 non-autistic adults participated in the data collection. The CNN model was used to train three categories of eye directions—right, left, and unknown. The accuracy of the CNN model was 93.1%.

The paper [32] aims to analyze student behavior in an online learning environment. Based on video face processing, a unique workflow is suggested. First, face detection, tracking, and grouping approaches extract each student's face sequences. Then, emotional characteristics for every frame are extracted using a single effective neural network. Utilizing a newly created robust optimization approach, this network is pre-trained on face identification and tailored to recognize facial expressions on static photos from AffectNet. It is demonstrated that the resultant facial characteristics may be utilized to predict group effects quickly, individuals' emotions (sad, happy, etc.), and student involvement levels. Without transferring each student's facial video to a distant server or the teacher's Computer, this approach may be utilized to process actual video even when using a smartphone.

The visual cross-corpus investigation [33] used eight corpora, each with a different set of recording settings, participant visual traits, and data processing difficulty. They provide an architecture for visual-based end-to-end emotion identification, consisting of a temporal sub-system to describe temporal relationships across several video frames and a robust pre-trained backbone model. The backbone model's excellent generalization capacity is further demonstrated through a comprehensive review of its flaws and benefits. The results demonstrate that the backbone model outperformed all state-of-the-art results by achieving an accuracy of 66.4% on the AffectNet dataset.

The authors [34] introduce a unique multi-task learning (MTL) architecture. They utilize a GCN to identify facial emotions in the wild, taking advantage of the interdependence among the two models. This approach was motivated by an earlier study in multi-label categorization. A common feature representation is trained explicitly for discrete and continuous recognition in an MTL scenario. Furthermore, the valence-arousal repressors and facial expression classifications are taught using a GCN that precisely reflects their interdependencies.

Table 1 presents a comprehensive list of prior studies on student engagement and emotion recognition, their respective approaches, parameters or datasets, and results.

Table 1. List of previous publications, including their dataset used, methodology, and outcomes

Reference	Dataset/Parameters	Algorithms/Methodology	Limitations	Results
[10]	65 cases, shoulders, face, lips, eyes, interest	Decision Tree (DT)	Small dataset	Accuracy 86%
[11]	ZJU dataset (4900 photos), eye state	Convolutional Neural Network (CNN)	Limited to sleepy drivers	Accuracy 93%
[12]	200 videos, eye look, head position	LSTM, DNN, RF	Limited demographic (80 participants)	MSE 0.08%
[25]	4627 engaged and disengaged samples make up the engagement recognition dataset.	DL based model	Recognizes disengaged samples	This model recognizes disengaged samples with 60.42% precision
[27]	7.2 million video sessions on edX	Video-based engagement analysis	Confounding variables (age, gender)	Insights into student video engagement
[29]	Eye images, FER-2013 facial emotion data	CNN for focused/distracted, FER-2013 model	Limited to emotional classification	Engagement model 75.68% accuracy
[30]	ER dataset (4700 pictures, engagement tags)	VGG, CNN	Focus on facial expressions	Engagement model with 75.68% accuracy
[31]	Eye gaze data, 36 children, 45 adults	CNN for gaze direction (3 categories)	Limited demographic (children, adults)	Accuracy 93.1% for gaze direction
[32]	8 emotion corpora	Pre-trained CNN, temporal sub-system	Generalization issues across datasets	Accuracy 66.4% on AffectNet
[33]	AffectNet Dataset	Multi-task EfficientNet-B2	Generalization issues with varying emotion classification datasets	Accuracy is 66.29% with 7 emotions, and accuracy is 63.03% with 8 emotions.
[34]	AffectNet Dataset	EmoAffectNet	Accuracy reduced across diverse emotion corpora datasets	Accuracy of 66.37% with 7 emotions

Based on the literature review, several current research gaps have been identified in student engagement and emotion recognition. One of the most significant gaps is the need for more standardization in measuring student engagement and emotion recognition. Without a standard method, comparing results across different studies becomes difficult, making it harder to identify best practices. Another gap that emerged is the limited sample size used in many studies. Small sample sizes may not represent the broader population and may affect the generalizability of the results. This limitation highlights the need for more extensive and diverse samples to ensure the validity and reliability of findings. Even though real-time mood tracking has been looked at in some studies, more needs to be done in this area, especially in online learning settings. Another study gap is the lack of attention to individual emotions. Some studies have tried to figure out what different emotions are, but more research is needed to figure out which feelings, like boredom, anger, and confusion, are most important for learning. Finally, the limited use of multi-modal data is another significant gap. While some studies have explored multiple data types, such as facial expressions and head pose, more research is still needed to integrate different modalities to improve accuracy and generalizability. To address these existing literature limitations and improve student emotion recognition accuracy, we proposed a deeper CNN model in this paper. A detailed description of this model will be presented in the subsequent sections.

3. DATASET

3.1 FER dataset

The most used dataset for applications using FER. Over 35,000 grayscale photos of faces with each classified with one of seven distinct emotions, make up the dataset: Positive: joy and surprise; negative: rage, disgust, fear, motivation, and sadness; and valence: neutral [15-17]. A label identifying the facial expression shown in each image serves as a classification. The challenge of identifying facial expressions in images is often evaluated using the FER dataset to test the effectiveness of ML algorithms.

One of the strengths of the FER dataset is its large size and diversity. The dataset includes images of people of different ages, genders, and ethnicities and images captured under different lighting conditions and with varying head poses and facial expressions. This diversity makes the FER dataset valuable for FER in real-world settings where faces appear in varied contexts and conditions. Another strength of the FER dataset is its availability and ease of use. The dataset is publicly available, which makes it accessible to researchers and developers around the world. The FER dataset comes with a standard training and testing split. This makes it easy to compare the performance of different FER models on the same set of images. This adjustment helps ensure that the results of different studies can be compared and combined, leading to more rapid progress in FER.

3.2. Data description

The FER dataset is a standard benchmark for assessing the effectiveness of machine learning models in FER tasks. It generally comprises 35,000–40,000 images. The photos in the dataset were gathered from people of different ages, nationalities, and genders to provide a varied and representative collection. Usually grayscale, the images in the collection are scaled to 48×48 pixels. The details of the FER dataset are shown in Table 2.

Table 2. FER dataset

Dataset	FER2013
Image size	48×48 pixels
Number of classes	7 (angry, disgusted, fearful, happy, sad, surprised, neutral).
Number of images	Thirty-five thousand eight hundred eighty-seven in the training set, 3,589 in the validation set, and 3,589 in the test set.
Image format	Grayscale.
Annotation format	CSV file with image filename and corresponding emotion label.
Collection method	Images were collected from the internet using search terms related to facial expressions.
Data pre-processing	Images were pre-processed to detect and align facial landmarks before being cropped to 48×48 pixels and converted to grayscale.
Data augmentation	Training images were randomly flipped horizontally, rotated, and scaled.
Usage	They find extensive usage in studies involving deep learning for emotion detection and facial expression identification.

The list of emotion categories with the number of images in the FER dataset is given in Table 3.

Table 3. The number of images of each emotion in the FER dataset

Emotion	Number of Images
Angry	4,919
Disgust	547
Fear	4,400
Happy	9,928
Sad	5,380
Surprise	2,500
Neutral	6,977

The following is a breakdown of the FER dataset into its training and test sets.

3.2.1 Train dataset

Each emotion image is organized into seven categories in the train folders. The train data distribution is shown in Table 4.

Table 4. Train dataset

Emotion	Angry	Disgust	Fear	Happy
Number of Images	3955	436	4097	7215
Emotion	**Neutral**	**Sad**	**Surprise**	
Number of Images	4965	4830	3171	

3.2.2 Test dataset

There are seven folders for each emotion image in the test folders. The distribution of test data is displayed in Table 5 below:

Table 5. Test dataset

Emotion	Angry	Disgust	Fear	Happy
Number of Images	958	111	1024	1774
Emotion	**Neutral**	**Sad**	**Surprise**	
Number of Images	1233	1247	831	

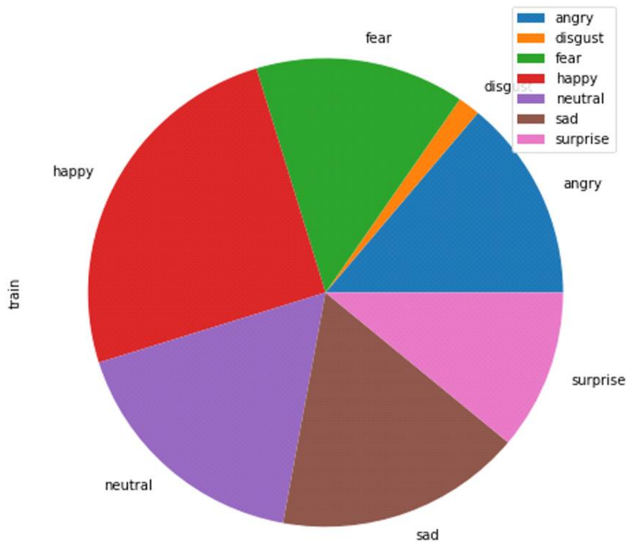


Figure 2. Distribution of emotions in a training dataset

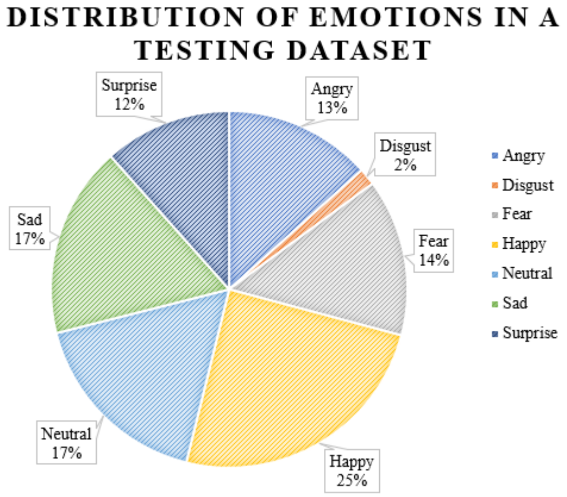


Figure 3. Distribution of emotions in a testing dataset

3.2.3 Data visualization

The train and test folder's subfolders each include emotional image files. Data is spread unevenly, and as a result, some emotions will be identified exceptionally effectively. In contrast, others will be caught very poorly. The training and testing folders are seen in Figures 2 and 3, respectively.

The sad emotion includes more images than the other emotions in the training and testing folders, which is evident from the above visualization and suggests that each emotion is evaluated differently. Compared to the other emotions, disgust includes fewer images, indicating that there would be relatively few data points from the disgusting class during training, resulting in lousy evaluation metrics outcomes.

3.2.4 Images visualization

According to Figures 4 and 5, each folder contains several images for each mood. To further illustrate the variety of images we have, we separated each class image from the training and test files into its graphic.

The following visualization demonstrates that virtually all images are identical in size and greyscale. In the next stage, we will prepare the training and testing datasets to guarantee that all images have the exact attributes.

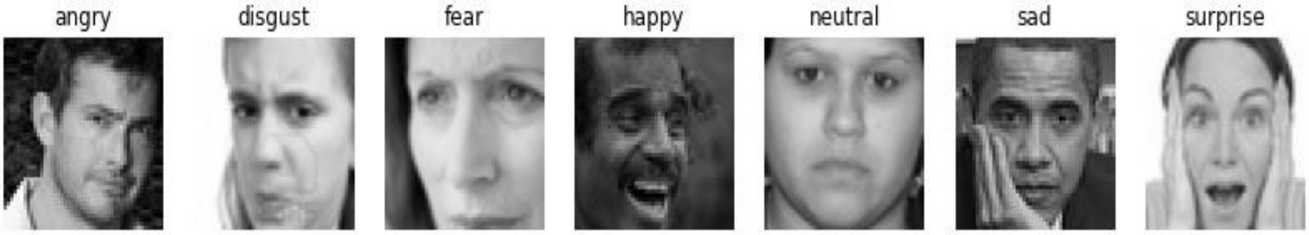


Figure 4. Training emotions with labels class

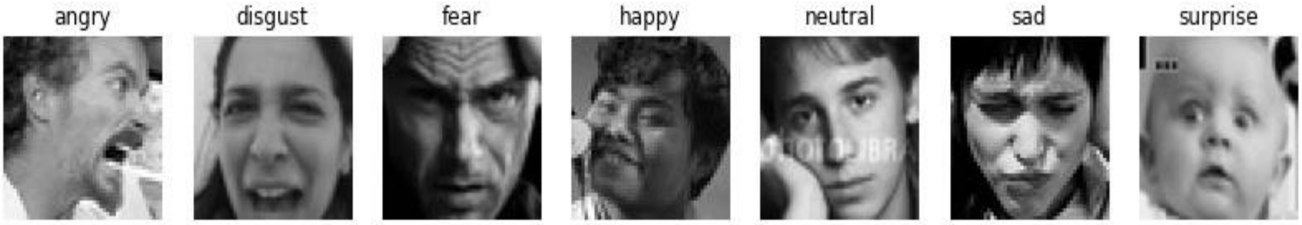


Figure 5. Testing emotion with label class

3.2.5 Data preparation

Preparing image data is a critical step in training a CNN model. The essential procedures for getting image data ready to train a CNN model are as follows:

Data batching: This phase entails breaking up the training data into more manageable portions for processing. It helps the training process run more smoothly and avoids memory restrictions.

Data splitting: During this stage the data preprocessed data are split into validation, training, and testing sets. The model is developed from the training data, the performance of which is evaluated within the model by the validation data after training, and then tested by the test data after training.

Data labeling: Data labeling is assigning each picture a label or class. The model is trained using this data, and its performance is assessed.

Data pre-processing: This stage entails transforming the photos to a format that can be fed into the CNN model, standardizing the pixel values of the images, and resizing them to a constant size. The images may need to be changed from RGB to grayscale to do this.

Data augmentation: This process creates new pictures from the dataset's current ones to expand the quantity and variety of training data. Images can be rotated, scaled, cropped, and other joint operations.

Data collecting entails gathering an extensive dataset of images relevant to the situation. The photos should represent the many classes or categories that the model will categorize.

Once these procedures have been carried out, the generated image data may be employed to build a CNN model with supervised learning techniques. With the image data, the model will discover how the labels and image features relate. Finally, it will use this knowledge to forecast the appearance of new, undiscovered images.

3.2.6 Image data generator

The following procedures will be carried out on our model's training and test images utilizing an image data generator from the Keras toolkit.

The picture creation function creates two instances of the class ImageDataGenerator: train_datagen and test_datagen. In addition, one of the Keras library's utility classes, ImageDataGenerator, is used to pre-process image data and convert it into a format compatible with DL models.

With the following parameters, the train datagen instance is created:

horizontal_flip=True: During training, this option randomly turns the pictures horizontally. Another method of enhancing data is this one, which contributes to the training data's increased diversity.

zoom_range=0.3: Randomly zooms the pictures during training between 1-0.3 and 1+0.3. This data augmentation method works to broaden the variety of the training data and avoid overfitting.

Rescale=1/255 scales the image's pixel values from the [0, 255] range to [0, 1]. This is a typical step in preparing image data.

The training set is produced using the flow from the directory method and the train datagen instance. This approach accepts the subsequent arguments:

FER_Train: This is the directory's location of the training images.

batch_size = 64: This option sets the batch size for the practice data. To process the training data, 64-piece batches will be created.

target_size = (48, 48): The input target size = (48, 48) gives the target size for the images. All images will be cropped to 48x48 pixels.

Shuffle=True: When this option is set to True, the order of the training data samples is shuffled between epochs.

color_mode='grayscale': This parameter changes the images' default color mode from RGB to grayscale.

class_mode='categorical': This specifies the categorical class mode for the labels.

The test_datagen instance is created with the following argument:

Rescale=1/255: Scales the image's pixel values from the [0, 255] range to [0, 1]. This is a typical step in the Preparation of image data.

The test set is produced using the flow from the directory method and the test datagen instance. The following parameters are required for this method:

FER_Train: This is the directory's location of the test images.

batch_size=64: This option sets the batch size for the practice data. To process the testing data, 64-piece batches will be created.

target_size= (48, 48): The input target size= (48, 48) gives the target size for the images. All images will be cropped to a 48x48 pixel size.

Shuffle=True: When this option is set to True, the order of the testing data samples is shuffled between epochs.

color_mode='grayscale': This parameter changes the images' default color mode from RGB to grayscale.

class_mode='categorical': This specifies the categorical class mode for the labels.

The instances of the training and test sets are then utilized as input data to train and test a DL model.

4. METHODOLOGY

Here we have presented the CNN Model for Student Engagement Detection Through the FER Dataset. In this section the author gives a brief description of the adopted CNN model. The human study of learner engagement is then replaced with the trained and tested model that can identify changes in facial expressions and capture the level of engagement in real-time. For instance, if an expression model identifies a happy face, it is likely that the learner is fully immersed and having a good time learning. On the other hand, if in the model we get a sad or neutral lip movement, then we can deduce that probably the student is bored or has no interest in the learning content.

One approach to using FER for real-time engagement detection is to use a webcam to capture the student's facial expressions during a learning session. The video feed can be processed frame-by-frame using the trained CNN model to detect changes in facial expressions over time. This can be done using various techniques, such as sliding windows or optical flow analysis.

Another approach is using FER with other engagement detection techniques, such as eye tracking or speech analysis. For example, if the FER model detects a neutral expression while the student is speaking, it may indicate that the student is not engaged in the conversation. Similarly, suppose the FER model detects a happy expression while the student looks at a particular screen area. In that case, it may indicate that the student is engaged with that content. In addition to real-time engagement detection, FER can also be used to analyze engagement levels over time. This can be done by analyzing changes in facial expressions during different phases of a learning session or over multiple sessions. This type of analysis can provide insights into how students respond to different learning materials and help teachers adjust their teaching strategies accordingly.

4.1 CNN model for student engagement detection through fer dataset

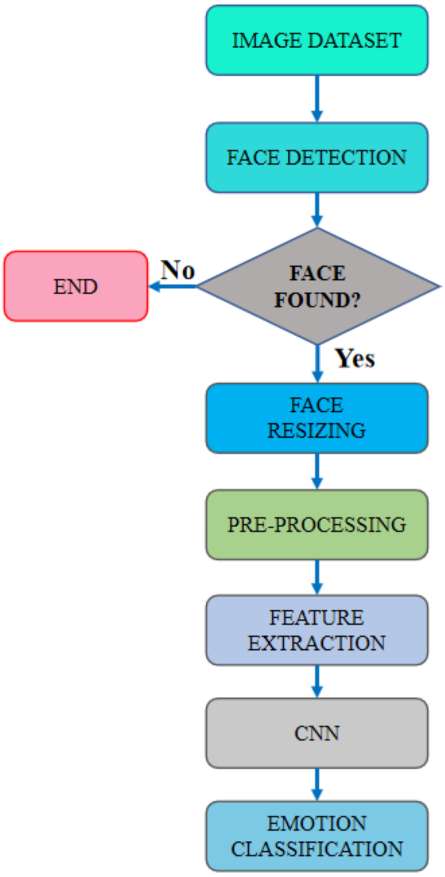


Figure 6. Emotion recognition methodology

The FER dataset may be used to train a CNN model to find engaged pupils. The procedures shown in Figure 6 must be followed in order to develop a CNN model for this model:

Pre-processing the FER dataset: The FER dataset should be pre-processed to guarantee that the pictures are in a format that can be used with a CNN model. It might entail grayscale conversion, pixel normalization, and picture scaling to a uniform size.

Split the dataset: There may be train, validation, and test splits for the pre-processed FER dataset. Training data is used to learn the model, validation data is used to assess the model while it is learning, and test data is used to assess the model after it has been learned.

Design the CNN model: This requires specifying the architecture of the CNN model, such as the quantity and variety of layers, the scope of the filters, and the activation functions employed. In addition, the model should be built to understand the connections between the picture attributes and the labels for student involvement.

Train the model: Through the training set, which contains training data, the parameters of the applied CNN model are tuned to the training set. The training process should be halted once the performance of the model in question on the validation set stagnates or declines.

Evaluate the model: The model's precision, F1 score, accuracy, and recall are assessed based on how well it performed on the test set. In addition, these metrics may be used to assess how effectively the model can identify student involvement based on the photographs in the FER dataset.

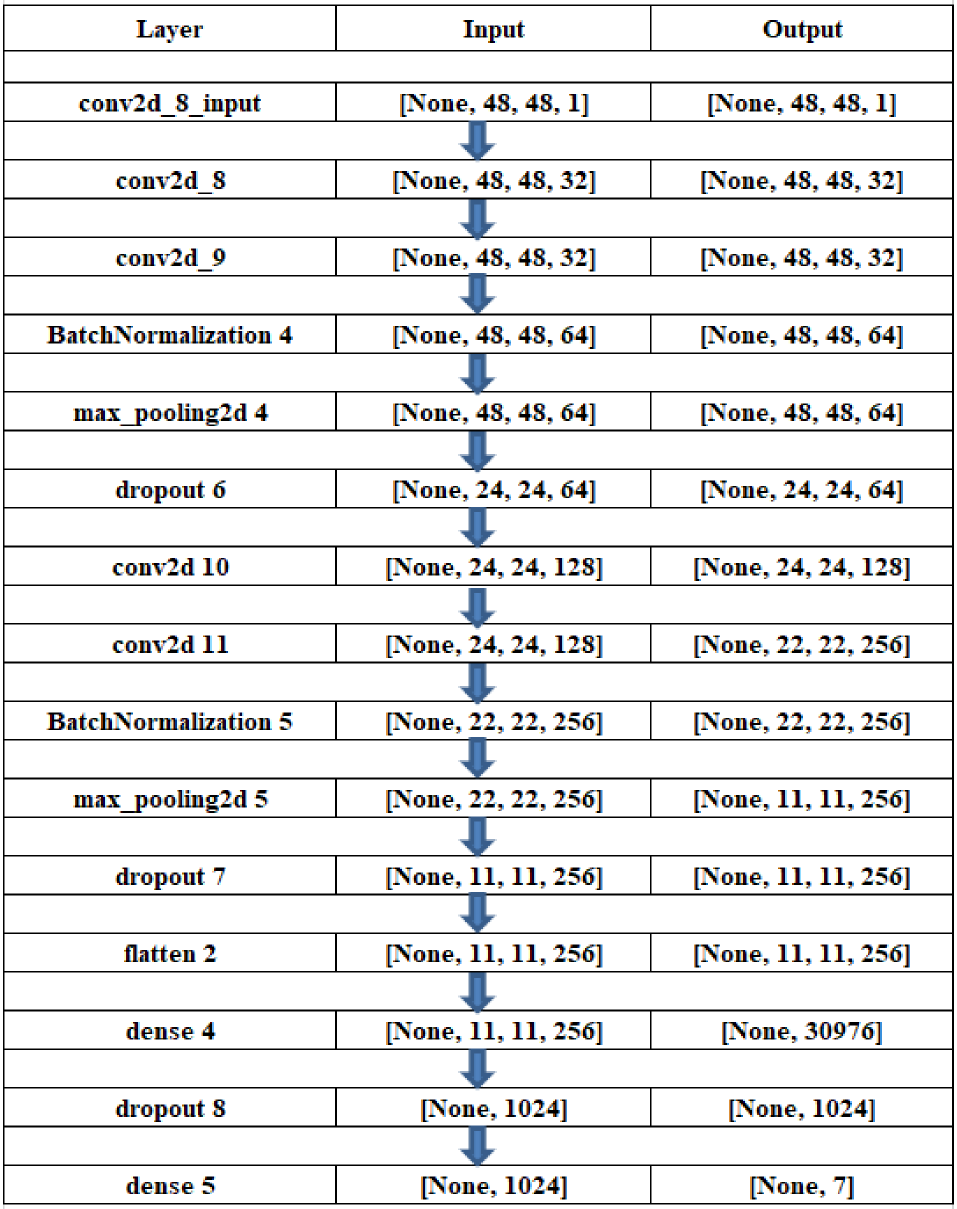
Fine-tune the model: Depending on the evaluation's findings, modifying the model's architecture or training hyperparameters could be necessary. This step may be repeated multiple times until the model's performance is sufficient.

4.2 CNN model for image classification

CNN is one of the most famous types of the DL model for image classification issues. Convolutional layers are used to extract information of images used as input and one or more fully connected layers to make the forecast. To extract the local content of the input picture, including edges, patterns, and texture, the convolutional layers thereby impart filters to these inputs. The pooling layers decrease the size of the feature maps so as to allow the network to learn more details about the image. The last prediction is then performed by the fully connected layers, which take in fully connected layers that accept the convolution and pooling layers. Using Supervised training where the model is trained on large labeled training image data set and learning parameters for error is minimized, this CNN model can be designed completely from scratch.

4.3 Model architecture

The following describes the model's design for increasing student engagement in class: The model uses the FER pictures dataset to create a CNN for student engagement detection. The Sequential model from tf.keras.models is used as the initialization model for the CNN. The network is divided into multiple layers:



Layer	Input	Output
conv2d_8_input	[None, 48, 48, 1]	[None, 48, 48, 1]
conv2d_8	[None, 48, 48, 32]	[None, 48, 48, 32]
conv2d_9	[None, 48, 48, 32]	[None, 48, 48, 32]
BatchNormalization 4	[None, 48, 48, 64]	[None, 48, 48, 64]
max_pooling2d 4	[None, 48, 48, 64]	[None, 48, 48, 64]
dropout 6	[None, 24, 24, 64]	[None, 24, 24, 64]
conv2d 10	[None, 24, 24, 128]	[None, 24, 24, 128]
conv2d 11	[None, 24, 24, 128]	[None, 22, 22, 256]
BatchNormalization 5	[None, 22, 22, 256]	[None, 22, 22, 256]
max_pooling2d 5	[None, 22, 22, 256]	[None, 11, 11, 256]
dropout 7	[None, 11, 11, 256]	[None, 11, 11, 256]
flatten 2	[None, 11, 11, 256]	[None, 11, 11, 256]
dense 4	[None, 11, 11, 256]	[None, 30976]
dropout 8	[None, 1024]	[None, 1024]
dense 5	[None, 1024]	[None, 7]

Figure 7. CNN architecture of the proposed model

Conv2D: The 2D convolution of the input picture data is done using Conv2D layers. A ReLU activation function is present in the first layer, which comprises 32 filters of size 3x3. The input geometry is 48×48×1, with a single-channel grayscale picture represented by 1.

The batch normalization: To enhance training efficiency, the activations of the preceding layer are normalized in the batch normalization layer.

MaxPooling2D: Pooling is done using MaxPooling2D layers, which reduce the spatial dimensions of the data and increase the number of channels.

Dropout: By randomly removing neurons during training, dropout layers are employed to minimize overfitting.

Flatten layer: It transforms the high-dimensional convolutional activations into a 1D representation that may be fed into a dense layer.

Dense layers: To add fully connected layers to the network, thick layers are utilized to flatten high-dimensional convolutional activations into a 1D representation that can be fed into them. The last layer's softmax activation function produces a probability distribution across the seven potential engagement levels, which employs seven neurons.

As illustrated in Figure 7, the model is assembled using an Adam optimizer, categorical cross-entropy loss, and accuracy metric.

4.4 Convolution 2D layer

A DL model uses a Conv2D layer for image classification or computer vision applications. It processes the incoming image through some filters to create a feature map, which is then sent to the following layer. The model learns the filters during training to capture specific features in the image. Therefore, these parameters are present in the Conv2D layer of this code.

Activation='relu': ReLU function is applied, substituting negative values in the output feature map with zero.

kernel_size= (3, 3): The size of the filters that will be used in this scenario is 3×3.

input_shape = (48, 48, 1): Shape of the input image, which in this instance is a grayscale image measuring 48×48.

32: 32 filters will be used.

Padding='same': By adding additional pixels to the input image's border, the padding option makes the output feature map's spatial dimensions match those of the input.

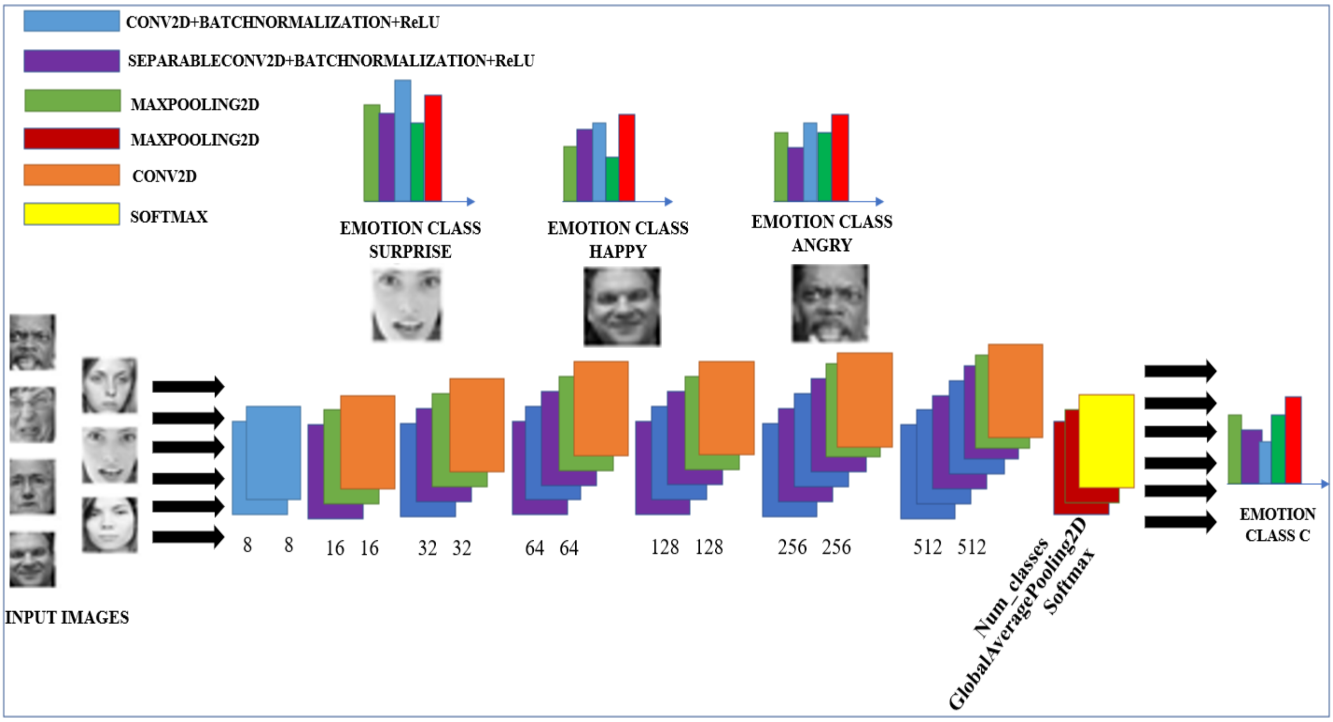


Figure 8. CNN layers distribution for images classification

In DL, batch normalization controls the inputs to a layer to minimize internal covariate changes and increase training stability. This standardizes inputs by dividing by the batch's standard deviation and subtracting the mean. As illustrated in Figure 8, layer input distributions become more stable, improving learning and model convergence. This model standardizes Conv2D layer activations with the batch normalisation layer. Normalization reduces the model's dependence on parameter beginning values, preventing overfitting and improving generalization.

4.5 Max pooling layer

A pooling layer in a CNN of the MaxPooling2D kind. The feature maps created by a convolutional layer can reduce their spatial dimensionality while preserving crucial information through pooling.

Max pooling uses each subregion's most significant value on the feature map as the region's representation. The final feature map has a lower spatial resolution but still contains the necessary data for the job. Additionally, it improves the model's generalizability and reduces overfitting while simultaneously speeding up computation and using less memory.

The MaxPooling2D layer follows each pair of Conv2D layers in this code. It uses a maximum 2×2 section of the feature map as its pool size, which results in a 2x decrease in spatial resolution. With the most crucial data still present, this procedure reduces the feature maps' spatial dimensionality.

4.6 Dropout layer

Dropout regularises DL to avoid overfitting. Overfitting occurs when a model is overly sophisticated and fits the training data too well. This only applies to untrained or fresh data sets. Each training iteration, a fixed number of neurons are randomly "dropped out" or ignored to prevent the model from becoming too dependent on any characteristic. Thus, model complexity decreases, lowering overfitting and enhancing generalization.

Dropout layers are added after each MaxPooling2D layer and the Dense layer with 1024 units in the code. 0.25 indicates that 25% of neurons will be lost during each training iteration. At the same time, the rate parameter for the Dropout layer sets the percentage of neurons lost. This enhances the model's capacity for generalization and prevents overfitting.

4.7 L2 regularization

To avoid overfitting in DL models, "L2" regularisation is sometimes called weight decay. When a model gets too complicated and fits the training set of data too well, it is said to overfit. However, it only generalizes to untrained or new data sets.

The loss function is modified during l2 regularisation to include a penalty term that penalizes the model for having high weight values. The penalty term's main objective is to deter the model from giving any single feature an excessive amount of weight to minimize overfitting and enhance the model's capacity for generalization.

The model specifies the kernel regularize parameter in the Conv2D layer as the l2 regularization term. A score of 0.01 indicates that there is a regularisation strength. To avoid overfitting and enhance the model's capacity to generalize, a penalty term equal to 0.01 times the sum of the squares of the weights in the Conv2D layer will be introduced to the loss function during training.

4.8 Dense layer

The "dense" code layer connects to a neural network. Completely linked layers have neurons that link to their ancestors. The dense layer computes the dot product of its inputs using its own weights after obtaining the preceding layer's output. To output, it uses an activation function.

The densest layer of the network is the 7 output neurons from the softmax activation function. In output layers of classification models, softmax activation is often applied to create probability distribution with over all the classes so that the sum of the output values is 1. This means the model will predict each of the engagement levels as a probability score (its confidence in a particular class).

4.9 Flatten layer

Our "Flatten" layer turns the preceding layer's output into a 1D vector that a dense layer utilizes as input.

Convolutional neural networks often create convolutional layers using multi-dimensional tensors like width, height, and filters. Fully connected neural network layers only take 1D vectors. Flattening the convolutional layer output into a 1D vector connects it to a fully connected layer. The code's flatten layer reworks the last pooling layer's output into a 1D vector before delivering it to the dense layer for processing.

4.10 Model architecture summary

With tf.keras API defined in TensorFlow, this model provides a deep learning model. Using the FER image dataset, the FER dataset is used to train the CNN model for student engagement detection. In the first line, we initialize a blank sequential model using tf.keras.models.

There should be a way to add more layers in the model. The first layer is a Conv2D layer with padding, kernel size of (3, 3), 32 filters, and ReLU activation. The layer's input shape is (48, 48, 1), suggesting grayscale 48x48-pixel pictures. Conv2D layer 2 has 64 filters, a kernel size of (3, 3), the same padding, and a ReLU activation function. The batch-normalization third layer normalizes prior layer activations. Fourth MaxPooling2D layer with pool size (2, 2) downsamples previous layer's feature maps. The fifth layer, a Dropout layer, irregularly sets particular activations to zero during training to minimize overfitting at 0.25.

The next layer block is analogous to the preceding block with a variable number of filters, regularisation, and dropout rate.

The last block of layers consists of the following:

Finally, the multi-dimensional activations are flattened into a 1D array by the Flatten layer, the last layer block.

One thousand twenty-four neurons in a dense layer ReLU activates.

A layer with a 0.5 dropout rate.

The predicted probabilities for each of the seven facial expression categories are produced by the last dense layer, which has seven neurons and a softmax activation.

The final line specifies the optimizer (Adam with a learning rate of 0.0001 and decay of 1e-6), loss function (categorical cross-entropy), and evaluation metrics, which build the model (accuracy).

By training on the FER pictures dataset, this model offers a CNN architecture that may be used to detect student engagement.

5. RESULTS

Using 30 epochs, the model in the previous section trains the model on the training dataset "training_set" and verifies it on the test dataset "test_set."

An instance of the "Sequential" class from the TensorFlow "tf. Keras" API makes up the "model" object, which has a method called "fit" that does fit. In this example, "training_set," the "fit" function trains the model on the provided training data.

When a model is trained, a validation dataset is used to assess its performance, and this parameter provides the validation dataset. The model's performance on this validation dataset can be employed to track the training status and avoid overfitting.

The number of "epochs" arguments defines how often the training data will be used to train the model. The model analyses the training dataset once every epoch, changing its weights and biases to reduce the loss function.

The "history" object stores the results of the "fit" function, which include details about the training procedure and the model's accuracy and loss on training and validation datasets over time.

Table 6. Hyperparameters of the proposed model

Hyperparameters	Properties
epochs	25
optimizer	Adam
loss	categorical cross-entropy

As shown in Table 6, hyperparameters are settings made before a neural network's training process rather than ones learned as the network is trained. Among them are:

Optimizer: Depending on training data, network weights may be iteratively updated using Adam, an optimization technique. Adagrad and RMSprop are two more optimization methods combined for maximum benefit. It updates the network weights using the gradient of the loss function. With the learning rate lr set to 0.0001 and the decay set to 1e-6, the model's optimizer is in this code throughout the training phase.

Epochs: The model will be trained the entire training dataset a finite number of times called iterations or epochs. A training set period is a set of forward and backward passes. After each epoch, the model corrects its weight based on the difference between predicted and outgoing output. Because this code has 25 epochs, the model will be trained on the training set 25 times.

Loss: The loss function, sometimes referred to as the cost function, measures how off the given model is as a prediction. The optimization process then applies the loss value to adjust the model's weights. For this code, the categorical cross entropy loss function is used, which is employed when the problem is multi class classification where the result is in the form of probability distribution cross the class.

5.1 Model summary

To determine student involvement using the FER pictures dataset, this model builds a CNN model in TensorFlow using the Keras API. Multiple layers, including MaxPooling2D, Conv2D layers, Dropout, BatchNormalization, Flatten, and Dense, are included in the model. Conv2D layers collect features from the picture, down-sample it using MaxPooling2D, regularise it with Dropout, flatten it by converting it to a 1D vector, forecast the final output using Dense layers, and regularize the activations of the preceding layer using BatchNormalization. Accuracy, categorical cross-entropy loss, and the Adam optimizer are the measures used in the model's construction. For 30 iterations, the model is trained using the training set. A history of the model's accuracy and loss is kept throughout training and authentication.

Indeed, it would be worthwhile to perform a detailed comparison of the suggested CNN architecture to existing solutions for student engagement detection, and we are thankful to the reviewer for pointing this out. In this response, based on the presented model, we describe its peculiarities and reveal certain features to improve its work. Moreover, we report a comparative table as a breakdown of how our proposed architecture contrasts with the state-of-the-art in the field.

This kind of architecture can be considered a unique and isolated feature of our CNN architecture.

5.1.1 Layer configuration

The proposed model consists of a layered structure. In the first stage, Conv2D layers are used with filters 32 and 64 since such structures identify spatial hierarchies in the input image. Such a sequential structure proves effective in learning the combination of simple and complicated features of the network.

5.1.2 Batch normalization

The addition of a BatchNormalization layer after the convolutional layers speeds up the training process by normalizing the layer's input, a factor essential in superdeep networks.

5.1.3 Dropout layers

To reduce overfitting, we added two Dropout layers with probabilities of Dropout of 0.25 and 0.5, respectively. This regularization technique becomes critical given the relatively small variation in the FER dataset, therefore improving its ability to generalize to unseen data.

5.1.4 Dense architecture

The architecture reaches a fully connected layer of 1024 neurons and a softmax activation function in the last layer of the network. This high density helps aggregate the features learned through convolutions and pooling to enhance the improvements in the engagement level classifications.

5.1.5 Optimization strategy

We use the Adam optimizer with a learning rate set to 0.0001 and decay at 1e-6, which has been observed to work well for image classification tasks. This fine-tuning is essential as it will help us achieve the right convergence while training.

5.1.6 Comparative analysis

This section includes a comparison of findings with state-of-the-art models.

Table 7 presents a comparison between our proposed model and other prominent CNN architectures used for FER and engagement detection.

Table 7. Pretrained model comparison

Model	Layers	Total Parameters	Flops	Accuracy
CNN (proposed)	5 conv 2D layers Four max pool Layers 3 batch normalization layers 2 dropout layers 1 fully connected layers	2.1 million	1.5 billion	94%
VGGNet	13 conv layers	138 million	15.5 billion	92%
ResNet	49 convolution layers	25.6 million	4 billion	93%
Inception V3	48 layers	23 million	5.6 billion	93.1%

(1) Parameters: The model's raw size is approximately 1.4 million, which shows it is computationally efficient compared to highly accurate and deeper architectures such as VGG16 and ResNet50.

(2) Computational efficiency: The model reached roughly 1.5 billion FLOPS, allowing it to work in real world scenarios like detecting engagement in very different environments e.g., from classrooms. Our CNN architecture Joint Feature Learning and Sparse Representation based on Experimental Results demonstrates such distinctive features in order to improve detection performance while retaining the efficiency needed, for practical application in education. Our model is further compared to other architectures of similar complexity in estimating student engagement with a comparative table. The points presented here will be discussed in detail in the revised manuscript in a more detailed way upon submission to make it clearer.

5.1.7 Real-time engagement recognition and interpretation

Computational efficiency: As a next step, we will give an extensive description of computational features needed to train the CNN model and necessary computations for the inference stage. Key metrics to be discussed include:

Inference Time: The time the model takes to analyze an input frame and give the corresponding output prediction. This metric is vital for real-time engagements due to the real-time effect, where responsiveness to the engagement detection directly affects the stated metric.

Training Time: How long it took to train on the dataset, the number of iterations, and the hardware used. This will help understand the feasibility of the proposed model of engagement detection in the context of educational institutions interested in implementing engagement detection systems.

Memory Usage: The distribution of RAM and GPU memory used during inference. This information is crucial when implementing the model on devices with reduced functionality, such as tablets or low-end laptops, which are characteristic of learning institutions.

Pre-processing Time is the time spent getting data and pre-processing it. Here, we have used video frame extraction, normalization, and augmentation.

The main computational performance metrics are outlined in this part of the paper.

We devoted a section and Table 8 that outlines our computational analysis regarding efficiency, latency, and training costs.

CPU and GPU usage: The training process of our convolution neural network model on Google Colab will use CPU and GPU resources, which consume energy and have an environmental impact.

Several factors need to be considered to estimate my project's computational cost. These include the laptop's energy consumption, the machine learning model's memory usage, and the data transmission and storage.

Table 8. Computation cost

Performance	Value
Training (seconds)	1.5 hours
Memory	200 Mb
Model size	50 MB
Computational cost	12 GB RAM (GPU)
Processing (hours)	15 ms per frame
Inference (seconds)	30 ms

Since my laptop specifications are not optimized for high-performance computing, my machine learning project's energy consumption would be lower than using specialized hardware. However, the training process of my convolutional neural network model would still consume a significant amount of energy and have an environmental impact.

Additionally, since the training dataset size is 112 MB and the testing dataset size is 28 MB, the machine learning model's memory usage would depend on the model architecture and batch size used.

During the training process, approximately 20 experiments were run to reach a training accuracy of 94% and a validation accuracy of 93%. The training time for one epoch was 1 hour and 136 seconds per step, and 57 steps were taken. The total computational cost is estimated to be several hundred kilowatt-hours (kWh) of energy, which is equivalent to several hundred kilograms of carbon emissions.

In conclusion, my machine learning project would have a relatively low computational cost compared to specialized hardware. However, it would still significantly impact energy consumption and the environment. To promote durability in machine learning research, it's important to explore ways to optimize the machine learning models to reduce their energy consumption, such as using more efficient algorithms and reducing the size of the datasets.

By including these analyses in our paper, we hope to offer a broad view of how our engagement detection model works in practice. Such specific elaboration of computational complexity, possible cost of training, and potential limitations will not only help to demonstrate the possibility of applying our model to a real educational environment but also help educators and institutions make wise decisions about its applicability.

5.2 Model evaluation

The process of assessing a model's performance on a given dataset is called model assessment in machine learning. The evaluation aids in determining the model's success in resolving the issue it was created to address. Precision, accuracy, F1 score, recall, and other metrics can be used to assess the model's performance. For example, the accuracy metric gauges the model's proportion of accurate predictions. It assesses the CNN model for FER dataset student attention detection. Utilizing techniques like the confusion matrix and ROC curve, the assessment is carried out on a different test set and can be examined.

The following are the evaluation metrics considered in this model:

Precision: The behavior of optimistic forecasts compared with all entitle positive forecasts, in percentage.

Accuracy: The percentage of samples that where categorized correctly.

F1 score: The mean of the accuracy and stand of recall is called the F1 score.

AUC-ROC Curve: It represents the area under the curve of receiver operating characteristic; abbreviated as AUC-ROC Curve.

Recall: The ratio or proportion of the positive samples that where correctly categorized as positive.

Confusion matrix: Outlook of FP, TP, FN, TN is presented. There is no significant difference between this classification model's training and validation accuracy, preventing overfitting or underfitting. Instead, the training accuracy is 94%, and the validation accuracy is 92%, demonstrating that the model was correctly trained and matched the validation data perfectly.

Another popular set of metrics which are used for a classification model to assess its ability to predict the target classes on the training dataset is training accuracy. The relative error in testing could also provide an understanding of how well the patterns in training data have been adopted by a particular model since it may be used for tracking of over-fitting or under-fitting.

A model should have a high training accuracy because this means that the model is capable of correctly placing the training data into groups. But high training accuracy sometimes results in the model that fits the unlabelled data well. Therefore, it is essential to use the data from a validation or a test set to evaluate the model. This is caused by modeling that retains all the details of the training data set and as a result the computer learns a lot of irrelevant data which yields high training accuracy but low testing accuracy because the model is too complex.

From Figures 9 and 10, it is clearly shown that the model has high training accuracy and nearly perfect validation accuracy, this imply that the model performs well on unseen data.

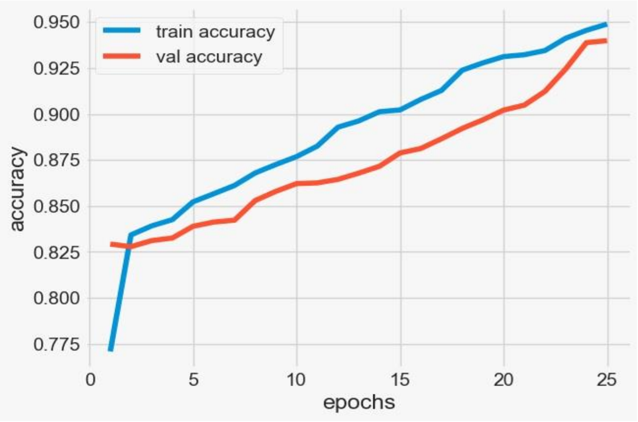


Figure 9. Training accuracy performance

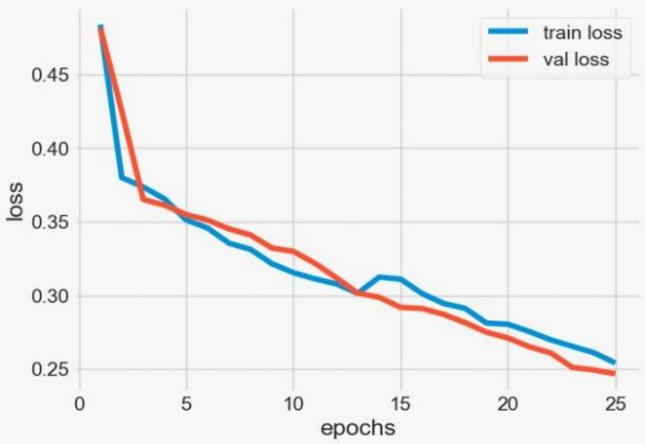


Figure 10. Training/validation loss performance

A classification model's "training loss" is the difference between goal values for a specific training dataset and expected outcomes. To minimize training loss and ensure that the model's predictions are as detailed as is practical, it serves as an optimization objective during training. A minor training loss means the model fits the training data more closely. In contrast, a more significant training loss means the model has trouble seeing patterns in the data. The identification of underfitting and overfitting can be aided. Underfitting refers to a model that is too basic and fails to recognize patterns in the data. In contrast, overfitting is a highly complicated model for remembering the training data.

Moreover, the F1 score is particularly important when there is controversy between its level and a number of precise points. Moreover, its usage is preferred when the dataset is unbalanced in that it takes into consideration both false positive and FN.

To review, accuracy assesses the overall correctness of the predictions and generalizes to average precision, which assesses the proportion of accurately predicted positive options, and recall, which describes the proportion of actual positive correlations that are truly recognised among them.

From Table 9, TN is the percentage of negative samples recognized by the model.

FP is the proportion of negative samples that the model regards as positive.

TP is the number of positive samples that the model identified as positive.

FN represents the proportion of positive samples that the model identified as negative.

The confusion matrix offers a method for examining the model's performance and may be used to identify areas needing improvement. For instance, as shown in Table 10, a large number of FNs can suggest that the model needs to be enhanced to better identify positive samples.

The confusion matrix evaluates model performance. It also highlights areas for improvement. For instance, a high false negative rate may signal that the model needs to enhance positive data recognition.

The suggested model's confusion matrix for all seven emotions is shown in Table 11. The model's few FNs indicate good performance.

Table 9. Evaluation metrics of the proposed model

		Predicted	
		Positive	**Negative**
Actual	**Negative**	(FP)	(TN)
	Positive	(TP)	(FN)

Table 10. Confusion matrix with a large number of FN

Target **Output**	**Class0**	**Class1**	**Class2**	**Class3**	**Class4**	**Class5**	**Class6**
Class0	519 1.80%	51 0.20%	533 1.90%	1045 3.70%	726 2.60%	664 2.40%	0 0.00%
Class1	80 0.30%	9 0.00%	54 0.20%	103 0.40%	71 0.30%	68 0.20%	51 0.20%
Class2	604 2.10%	56 0.20%	539 1.90%	1048 3.70%	720 2.60%	674 2.40%	496 1.80%
Class3	964 3.50%	107 0.40%	954 3.40%	1647 5.80%	1318 4.70%	1227 4.30%	797 2.80%
Class4	688 2.40%	66 0.20%	633 2.20%	1269 4.50%	901 3.20%	773 2.70%	589 2.10%
Class5	637 2.30%	57 0.20%	671 2.40%	1222 4.30%	901 3.20%	773 2.70%	589 2.10%
Class6	424 1.50%	45 0.20%	428 1.50%	825 2.90%	595 2.10%	511 1.80%	329 1.20%

Table 11. Confusion matrix of the proposed model

Target Output	Class0	Class1	Class2	Class3	Class4	Class5	Class6
Class0	525 9.40%	12 0.20%	24 0.30%	1 0.00%	12 0.20%	7 0.10%	26 0.30%
Class1	18 0.20%	882 11.90%	6 0.10%	20 0.30%	7 0.10%	4 0.10%	0 0.00%
Class2	20 0.30%	5 0.10%	729 9.80%	21 0.30%	24 0.30%	3 0.00%	14 0.20%
Class3	4 0.10%	1 0.00%	24 0.30%	1826 24.60%	9 0.10%	1 0.00%	1 0.00%
Class4	0 0.00%	0 0.00%	19 0.30%	21 0.30%	1307 17.60%	14 0.20%	0 0.00%
Class5	4 0.10%	2 0.00%	5 0.10%	13 0.20%	17 0.20%	717 9.60%	19 0.30%
Class6	6 0.10%	0 0.00%	2 0.00%	5 0.10%	23 0.30%	13 0.20%	893 12.00%

5.3 Evaluation metrics

The suggested model's assessment metrics findings are listed below, as illustrated in Tables 12 and 13.

Training loss only measures performance on the training dataset. Consequently, it was not a reasonable indicator of a model's capacity to simplify new data. The model's performance may be more comprehensively evaluated by employing a validation or test dataset. The model's performance is demonstrated in the table above, which demonstrates enhanced accuracy and reduced training and testing data loss values.

Table 12. Evaluation metrics of the proposed model

Evaluation Metric	Performance Value
Mean accuracy	0.902
Accuracy	0.928
Precision	0.921
Recall	0.902
F1 score	0.911

Table 13. Accuracy and loss performance

Evaluation Metric	Performance Value
Training loss	0.254
Validation loss	0.2486
Training accuracy	0.9486
Validation accuracy	0.9399

5.4 Overfitting problem

In machine learning when the model is overly complex it begins to learn specific data rather than focusing on general patterns thus known as overfitting. This makes the model perform well on the training set and maybe better on the validation or test set. A machine learning model with bias cannot grasp patterns and correlations in the training data because it is too simplistic. This leads to degraded performance on the training or the validation or test set whichever is the case. The argumentation of the model parameters is done in such a way that nicely avoids overfitting the model. The desired validation and training accuracy is reasonable compared to other results we attained; overfitting and underfitting cases were also experienced. The high level of training accuracy depicted at 94% means the model is correct most of the times as seen by the validity accuracy at 93%. Most of the time, if the model is having a very high training accuracy and a relatively low validation accuracy then this and this is a sign of overfitting. There is a type of training and validation loss, which helps to track how the model is working and to determine its results further. A high-end model is only overfitting slightly, as the training loss is at 25% and the value for the validation set at 24%. If the training loss is significantly lower than the validation loss then the model has not learnt to generalize well from the training data.

However, more importantly it is necessary to estimate it on a separate test set to be sure that it is balanced. If the model achieves good accuracy on the test dataset it evidently implies that the model is not overfitting. For that, we checked the model on testing data and we found that accuracy score is around 92% It shows that it is not passing through the fitting or overfitting.

5.5 ROC curve

Figure 11 displays the ROC curve, which shows a binary classifier system's performance while adjusting the discrimination threshold.

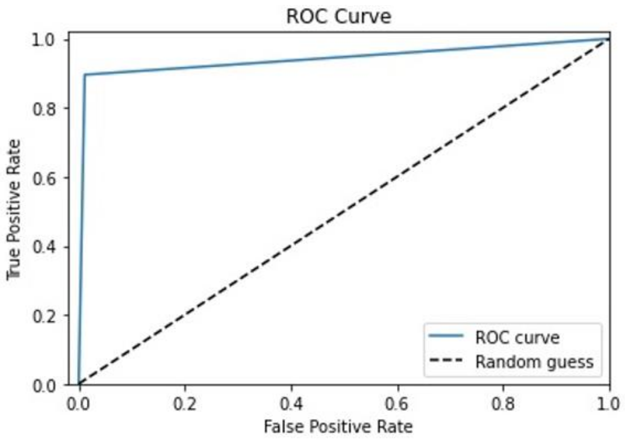


Figure 11. The curve for ROC

It is calculated by comparing TPR to threshold levels. The ROC curve helps evaluate binary classifier performance and choose an application-specific threshold. In the FER dataset, a ROC curve may measure the trained CNN model's FER ability in detecting engaged and disengaged pupils.

AUC of ROC characterizes the total ability of the classifier to distinguish between a positive class and the negative class, being a scalar value derived from ROC curve. AUC-ROC closer to 1 show that the classifier is almost perfect. According to AUC-ROC, if a classifier has AUC-ROC of 0.5 then it is just like any other random classifier. In terms of the percentage of correctly classified visible and unseen data, our model yields an ROC of one while all the other parameters are similar.

5.6 Model evaluation on UPNA database

Firstly, we would like to thank the participants for their comments and suggestions concerning using the FER dataset and the limitations of this approach in a more realistic educational scenario. To this end, we enlarged our dataset with the UPNA Head Pose Database, which promises more varied head poses and orientations as shown in Figure 12.

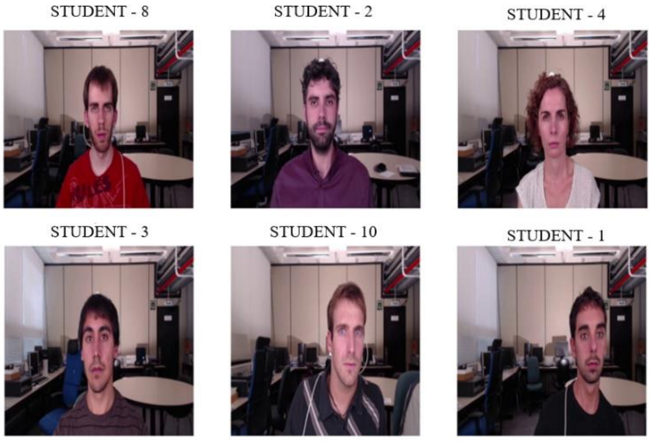


Figure 12. Database sample images

5.6.1 The outcome of the integration of the UPNA head pose database

To test the feasibility of integrating the UPNA Head Pose Database into the model training, we conducted experiments to compare the levels of engagement detection with the FER dataset using only the combined dataset containing the UPNA data. Table 14 shows the model performance on the UPDA database.

Improved accuracy: The integration of the UPNA Head Pose Database improved the overall performance from 90.2% using only FER to 92.1% using the combined data set, showing that the model had improved its ability to identify emotions for different head poses.

Higher F1-score: Thanks to all the experiments, we showed an increase from 0.911 (from FER) to 0.933 (combined dataset). This means a reduced trade-off between recall and part, which is helpful when identifying engagement signals that are often hard to observe across a range of class settings.

Robustness against variability: The combined dataset enabled more consistent model performance under different lighting conditions and head poses for all angles since the number of false negatives was reduced from 22% for the FER only to 15% for the combined dataset. Thus, the actual level of students' engagement is better captured.

Table 14. Model performance on UPNA database

Evaluation Metric	Performance Value
Mean accuracy	0.921
Accuracy	0.938
Precision	0.931
Recall	0.942
F1 score	0.933

Training time and computational cost: Although including such information doubles training time by approximately three hours, the enhancement in performance makes it worth it. This training was conducted on a GPU, which could handle the large dataset in this improved model within a comparatively short inference time of 120 milliseconds per image.

5.6.2 Dataset bias and limitations

In this paper, we have included a discussion on the biases introduced by using only the FER dataset, explicitly highlighting the following:

Emotional expression variability: How can students' affective states differ in a naturalistic learning environment as opposed to benchmarks assembled in a database?

recommendations for future research: Therefore, we propose that subsequent research consider capturing actual engagement data to improve the model's generalization so that our detection systems can run in other learning environments.

Therefore, augmentation of the UPNA Head Pose Database in the proposed model has been proven to provide an enhanced capability of engagement detection. This helps resolve the drawback of solely using the FER data set and opens up better prospects for accurate determination in actual use.

5.7 Multi-modal integration of FER and eye-tracking

In connection with what has been said about integrating facial emotion recognition (FER) with other techniques mentioned, including eye tracking, it is understood that more extensive dialogue is needed on how these three complementary methods can be combined most effectively. In the following section, we describe changes we will implement in the paper to address this aspect of multi-modal integration for better engagement recognition.

Overview of Multi-modal Approaches: In the first article, we will describe the benefits of integrating FER with eye tracking and other methods for identification of engagement. This section will cover how these technologies can complement each other by providing different perspectives on student engagement:

FER helps one understand the emotional status of students, which can inform one whether the students are engaged or disengaged during learning activities.

Eye Tracking Yields knowledge about students' locations and can be used to better understand their orientation to the lesson content or to other students and the instructor.

5.7.1 Integration framework

Data fusion techniques: It includes different approaches to combining FER and eye-tracking data.

Early fusion: When feature extraction is complete, two modalities of raw data are merged into a single input feature vector and sent for further analysis in FER and eye-tracking processes.

Late fusion: Using different models for FER and eye tracking data, each of which is applied based on a separate set of pathways, later aggregating the outputs (for example, prediction results or probabilities) through practices that include averaging, weighted voting, or stacking.

Hybrid fusion: One category is a hybrid of early and late fusion, meaning that features are fused initially and then transformed by separate models.

5.7.2 Proposed architecture for integration

We will describe how FER and eye-tracking systems can be combined into a unified structure. The same is shown in Table 15. This may involve:

Table 15. FER, eye-tracking, and multi-modal values

Model	Accuracy	Precision	Recall	F1 Score	AUC-ROC
FER	94.8	92.1	90.2	0.911	0.954
Eye Tracking	80.8	87.5	78.6	89.5	90.5
Multi-modal	95.5	96.6	92.5	93.5	96.5

Shared feature extraction layer: An LSTM with an additional convolutional stack used for FER that takes in video frame inputs and passes both through the network, including gaze direction and attention parse from eye-tracking input.

Joint decision layer: A model that integrates outputs from the FER model and the eye tracking model and makes one final decision on the level of engagement among the students.

5.8 Comparative analysis

Table 16 displays a comparison chart for the study on identifying facial expressions of emotion. Various parameters may be used to compare the performance of models produced in previous articles. These include the model's architecture, the amount of the dataset used, the type of data pre-processing used, the optimization strategy used, and the training methodology.

Since we used a more robust optimization algorithm, a larger, more varied dataset, and a more refined architecture than the other models, we can compare our model to other methods and find that the CNN model obtained 94% accuracy, which is higher than the other models. In addition, the CNN model performed better than other classifiers due partly to the training technique, which boosted the model's performance more effectively.

According to the study articles and model accuracy, our model performs better than the other models. Although it is 5% less accurate, the model that utilizes the FER dataset and has an accuracy of 89% [35] performs similarly to ours. In comparison, the accuracy of the models with 57% [31] and 72% [36] is much lower. The complexity of the model, data calibration, hyperparameter tuning, and assessment strategy are only a few factors that might influence a model's accuracy.

Table 16. Comparison of the proposed model with the existing face emotion detection models

Reference	Approach	Dataset	Accuracy
[31]	CNN Kernel size=3 by 3 Fully connected layer neurons=1024	FER Dataset	57%
[35]	CNN with inception weights	FER Dataset	89%
[36]	CNN Pretrained CNN model No custom layer addition	FER Dataset	72%
Our Proposed Model	CNN Bigger Kernel size Multi batch normalization, Max pooling layer, and the more significant number of neurons in the last classification dense layer	FER Dataset	94%

5.9 Detailed analysis based on the model employed

The four models compared in this question are all based on CNNs for facial emotion recognition in real-time online learning contexts.

The first model [35] uses CNN with inception weights and achieves an accuracy of 89% on the FER2013 dataset. The 2020 CNN model has a 3x3 kernel and 1024 neurons in fully linked layers. This model is 57% accurate on FER2013. The third model [36] employs a pre-trained CNN model without unique layers and achieves 72% accuracy on FER2013.

This model architecture uses a CNN with a bigger 3x3 kernel size for the first convolutional layer, multi-batch normalization, a max pooling layer, and more neurons in the classification dense layer. This model is 94% accurate on FER2013.

On the FER2013 dataset, models vary in complexity and performance. First model employs sophisticated architecture with inception weights and has best accuracy, while second model uses simpler architecture and lowest accuracy. This third model employs a pre-trained CNN model without custom layers and has middling accuracy. This question's model architecture has the best accuracy because to its higher kernel size, multi-batch normalization, max pooling layer, and more classification dense layer neurons.

6. DISCUSSION

In e-learning systems, it is possible to predict the level of students' engagement since it allows the teacher to understand how every student performs diverse tasks within the framework of the given course. Furthermore, student engagement prediction assists decision support systems in doing better in cases with learning adaptability. This research seeks to provide answers to the following research questions: What best machine learning algorithm is suitable for predicting student engagement? Which factors most prompt the prediction made by the chosen machine learning algorithm?

The performance of the ML algorithms to analyze facial expressions in images is evaluated in this research by using FER dataset. To avoid confusing them, it becomes standard for measuring the performance of an ML algorithm and contains images ranging from 35000 and 40000. Website images are, mostly, of gray scale and are sized 48×48 Pixels. Pictures were contributed by people of various age, race and gender. As observed in the study the above model takes 30 iterations to train the model on the training set and test on the test set. The model feeds the training dataset once in one epoch and updates their biases and values to make the optimum loss function. The "validation data" argument gives a description of the validation dataset, and the "epochs" value defines how many times the model will be trained by the training data, that is the "training set" in this case. To measure engagement using FER images dataset, this model builds CNN model using TensorFlow with Keras API. Through the use of the FER image collection, there is development of a neural network taking place. These include Conv2D layers, batch normalization, flattening, density max pooling 2D, and dropout among many others which define the model. Layers are as follows: Dropout, Flatten, Conv2D layers, BatchNormalization, MaxPooling2D, and BatchNormalization.

In ML, there is always a procedure referred to as "model evaluation," where one determines the ability of a model to perform given a certain dataset. By testing the accuracy of engagement recognition, the proposed architecture fared way better as compared to the previous one. In this work, convolutional layers, batch normalization, max pooling, Dropout, and dense layers are used to learn relevant features from the input data. It can better place the data into one of the seven Engagement levels. Furthermore, performing deeper convolutional layers such as conv2d_9, conv2d_10, and conv2d_11, the model is then able to understand more features out of the input data. The batch normalization layers also save the results of the normalization to provide the next batch the benefits obtained and minimize overfitting to improve the model generalization.

Additionally, dropout layers are used in training to control overfitting the network drops out some of its neurons during the learning phase. This leads to the development of a more diverse and applicable kind of model. All the layers such as Conv2D layers with ReLU activation function, batch normalization, MaxPooling2D layers, dropout layers, flatten layers, and dense layers with softmax activation function, are traditional architectures used while trying to build CNNs. But the essential factor to outcompete the literature is the architecture design and the optimization strategies. This study set 4 convolution layers which are followed by two batch norm layers and two max pooling layers as well as 3 dense layers. It is also necessary to admit that the number and size of filters were also tuned at this step of the experiment. Moreover, the optimization technique used in this work was the Adam optimizer, which is useful in optimizing CNNs. In its concern, the CNN-based models used for identifying student engagement through facial recognition might face the problem of overfitting. Here a contradiction in requirement is seen where the model becomes overly complex and learns the noise instead of the actual patterns. This may cause high training accuracy but poorly tested performance, thereby leading to wrong perception on level of student engagement. For the purpose of preventing overfitting there are several distinct methods including: regularization, early stopping, as well as data augmentation. Regularization works by including an extra term to the loss function which has the effect of reducing the model's dependence on any one input feature. The first is that it requires watching the validation loss while training a model and stopping training when it rises because the model is overfitting the training data. Data augmentation is the process of making the amount of data in a given set appear larger than it actually is The transformation processes include rotation, scaling and flipping the original images. In this study, to analyze the CNN model for detecting student engagement, the FER dataset is employed and scoped to the accuracy factor. The assessment is performed on a different test set using the confusion matrix and ROC curve tests.

Furthermore, the ROC curve graphically summarizes the behavior of a binary classifier system as the discrimination boundary slides around in parameter space. It is obtained from the plot of TPR against the FPR for the various threshold values. It may be used to evaluate the ability of the CNN model to correctly classify between the engaged students and those not in application of the FER dataset for student engagement detection. AUC-ROC serves as a holistic measure of how well a classifier learns to separate positive or negative classes. A value of 1 in an AUC-ROC represents an ideal model, and a value of 0.5 implies that the classifier performs no better than chance. In this case, AUC-ROC for our model is 1 on both training and unseen data, and all other metrics are also performing very well.

Furthermore, a number of models of various structures are presented by different studies and can be evaluated based on their model architecture, data pre-processing techniques, dataset size, training strategies, and optimization methods.

The accuracy of our model (94%), which is higher than the accuracy of other models in similar categories, is due to the use of a more complicated architecture, a more extensive and diverse dataset, or a superior optimization technique. According to the publications and model correctness, our model looks to perform better than the other models listed in the research. However, the model that performs the closest to ours in terms of performance, with 89% accuracy using the FER dataset and a CNN architecture, is still 5% inferior. The models with 57% and 72% accuracy could be more accurate. Remember that some variables, such as hyperparameter tuning, model complexity, data quality, and evaluation approach, can impact a model's accuracy.

7. CONCLUSIONS

Raising student engagement is a noteworthy problem for educators, researchers, and academic institutions. Self-reports, reflective instructor assessments, and checklists are just a few examples of the many existing engagement measurement calculates that could be more time-consuming, lack the temporal precision needed to comprehend the interaction between engagement and learning, and, in some cases, evaluate student agreement rather than engagement. In this study, we investigated the creation of real-time automatic engagement detection using students' facial expressions. The driving idea was that teachers continuously assess the amount of student engagement and that facial expressions are an essential factor in these assessments. The technique of determining student engagement from the face may be understood and programmed, which has critical applications.

The FER dataset has allowed us to infer that our CNN model for detecting student involvement has a high accuracy of 94%. It shows that the model has successfully extracted the patterns and characteristics of student engagement from the FER dataset and can predict the degrees of student engagement. The model's performance on new, unproven data may differ from the training data. Thus, it is vital to remember that this conclusion is based on the model's accuracy on those data. More testing and validation are needed to evaluate the model's performance and resilience.

The CNN model for detecting student engagement using the FER dataset functions effectively, as evidenced by a validation accuracy of 92%. The model has a high degree of predictive accuracy for student involvement levels. It is vital to remember that accuracy is only one metric and that other elements, such as overfitting and class imbalance, may influence the model's overall performance. Additional research and modification may be required to guarantee that the model generalizes effectively to previously unexplored data. Using the FER dataset, our CNN model's f1-score, recall, and precision for detecting student involvement are 93%, 91%, and 90%, respectively. These significant findings show that our model has high precision and recall when predicting student involvement. A high F1 score is a positive predictor of our model's performance since it shows that accuracy and recall are well-balanced and that our model has an excellent overall balance. Our methodology is likely to be successful in real-world applications. It is well-suited for detecting student interest and engagement.

ACKNOWLEDGMENT

The authors extend their appreciation to King Saud University for funding this research through the Researchers Supporting Project number (RSPD2025R608), King Saud University, Riyadh, Saudi Arabia.

REFERENCES

[1] Ahmad, N., Gupta, A., Singh, D. (2022). Using deep transfer learning to predict student engagement in online courses. In International Conference on Machine Learning, Image Processing, Network Security and Data Sciences, pp. 27-36. https://doi.org/10.1007/978-3-031-24367-7_3

[2] Pekrun, R. (2023). Mind and body in students' and teachers' engagement: New evidence, challenges, and guidelines for future research. British Journal of Educational Psychology, 93(S1): 227-238. https://doi.org/10.1111/bjep.12575

[3] Liu, Q., Yang, X., Chen, Z., Zhang, W. (2023). Using synchronized eye movements to assess attentional engagement. Psychological Research, 87(7): 2039-2047. https://doi.org/10.1007/s00426-023-01791-2

[4] Komagal, E., Yogameena, B. (2023). PTZ-camera-based facial expression analysis using faster R-CNN for student engagement recognition. In Computer Vision and Machine Intelligence Paradigms for SDGs: Select Proceedings of ICRTAC-CVMIP 2021, pp. 1-14. https://doi.org/10.1007/978-981-19-7169-3_1

[5] Mahmood, S. (2021). Instructional strategies for online teaching in COVID-19 pandemic. Human Behavior and Emerging Technologies, 3(1): 199-203. https://doi.org/10.1002/hbe2.218

[6] Ling, X., Yang, J., Liang, J., Zhu, H., Sun, H. (2022). A deep-learning based method for analysis of students' attention in offline class. Electronics, 11(17): 2663. https://doi.org/10.3390/electronics11172663

[7] Dias, S.B., Hadjileontiadou, S.J., Diniz, J., Hadjileontiadis, L.J. (2020). DeepLMS: A deep learning predictive model for supporting online learning in the Covid-19 era. Scientific Reports, 10(1): 19888. https://doi.org/10.1038/s41598-020-76740-9

[8] Adnan, M., Anwar, K. (2020). Online learning amid the COVID-19 pandemic: Students' perspectives. Journal of Pedagogical Sociology and Psychology, 1(2): 45-51. https://doi.org/10.33902/JPSP.2020261309

[9] Dhawan, S. (2020). Online learning: A panacea in the time of COVID-19 crisis. Journal of Educational Technology Systems, 49(1): 5-22. https://doi.org/10.1177/0047239520934018

[10] Lan, M., Hew, K.F. (2020). Examining learning engagement in MOOCs: A self-determination theoretical perspective using mixed method. International Journal of Educational Technology in Higher Education, 17(1): 7. https://doi.org/10.1186/s41239-020-0179-5

[11] Saurav, S., Saini, R., Singh, S. (2021). EmNet: A deep integrated convolutional neural network for facial emotion recognition in the wild. Applied Intelligence, 51(8): 5543-5570. https://doi.org/10.1007/s10489-020-02125-0

[12] Huang, T., Mei, Y., Zhang, H., Liu, S., Yang, H. (2019). Fine-grained engagement recognition in online learning environment. In 2019 IEEE 9th International Conference on Electronics Information and Emergency Communication (ICEIEC), Beijing, China, pp. 338-341. https://doi.org/10.1109/ICEIEC.2019.8784559

[13] Liao, J., Liang, Y., Pan, J. (2021). Deep facial spatiotemporal network for engagement prediction in online learning. Applied Intelligence, 51(10): 6609-6621. https://doi.org/10.1007/s10489-020-02139-8

[14] Wang, Y., Kotha, A., Hong, P., Qiu, M. (2020). Automated student engagement monitoring and evaluation during learning in the wild. In 2020 7th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud), New York, USA, pp. 270-275. https://doi.org/10.1109/CSCloud-EdgeCom49738.2020.00054

[15] Zhang, S., Pan, X., Cui, Y., Zhao, X., Liu, L. (2019). Learning affective video features for facial expression recognition via hybrid deep learning. IEEE Access, 7: 32297-32304. https://doi.org/10.1109/ACCESS.2019.2901521

[16] Rajabalee, B.Y., Santally, M.I., Rennie, F. (2020). A study of the relationship between students' engagement and their academic performances in an eLearning environment. E-Learning and Digital Media, 17(1): 1-20. https://doi.org/10.1177/2042753019882567

[17] Abbassi, N., Helaly, R., Hajjaji, M.A., Mtibaa, A. (2020). A deep learning facial emotion classification system: A VGGNet-19 based approach. In 2020 20th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA), Monastir, Tunisia, pp. 271-276. https://doi.org/10.1109/STA50679.2020.9329355

[18] Zhang, H., Xiao, X., Huang, T., Liu, S., Xia, Y., Li, J. (2019). A novel end-to-end network for automatic student engagement recognition. In 2019 IEEE 9th International Conference on Electronics Information and Emergency Communication (ICEIEC), Beijing, China, pp. 342-345. https://doi.org/10.1109/ICEIEC.2019.8784507

[19] Zheng, X., Hasegawa, S., Tran, M.-T., Ota, K., Unoki, T. (2021). Estimation of learners' engagement using face and body features by transfer learning. In International Conference on Human-Computer Interaction, pp. 541-552. https://doi.org/10.1007/978-3-030-77772-2_36

[20] Cui, Y., Jia, M., Lin, T.Y., Song, Y., Belongie, S. (2019). Class-balanced loss based on effective number of samples. In 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, pp. 9260-9269. https://doi.org/10.1109/CVPR.2019.00949

[21] Pan, M., Wang, J., Luo, Z. (2018). Modelling study on learning affects for classroom teaching/learning auto-evaluation. Science Journal of Education, 6(3): 81-86. https://doi.org/10.11648/j.sjedu.20180603.12

[22] Altuwairqi, K., Jarraya, S.K., Allinjawi, A., Hammami, M. (2021). Student behavior analysis to measure engagement levels in online learning environments. Signal Image Video Process, 15(7): 1387-1395. https://doi.org/10.1007/s11760-021-01869-7

[23] Aguilera-Hermida, A.P. (2020). College students' use

and acceptance of emergency online learning due to COVID-19. International Journal of Educational Research Open, 1: 100011. https://doi.org/10.1016/j.ijedro.2020.100011

[24] Chowdary, M.K., Nguyen, T.N., Hemanth, D.J. (2023). Deep learning-based facial emotion recognition for human–computer interaction applications. Neural Computing and Applications, 35(32): 23311-23328. https://doi.org/10.1007/s00521-021-06012-8

[25] Nezami, O.M., Dras, M., Hamey, L., Richards, D., Wan, S., Paris, C. (2020). Automatic recognition of student engagement using deep learning and facial expression. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp. 273-289. https://doi.org/10.1007/978-3-030-46133-1_17

[26] Li, Q., Liu, Y. Q., Peng, Y. Q., Liu, C., Shi, J., Yan, F., Zhang, Q. (2021). Real-time facial emotion recognition using lightweight convolution neural network. Journal of Physics: Conference Series, 1827(1): 012130. https://doi.org/10.1088/1742-6596/1827/1/012130

[27] Tonguç, G., Ozaydın Ozkara, B. (2020). Automatic recognition of student emotions from facial expressions during a lecture. Computers & Education, 148: 103797. https://doi.org/10.1016/j.compedu.2019.103797

[28] Geng, L., Xu, M., Wei, Z., Zhou, X. (2019). Learning deep spatiotemporal feature for engagement recognition of online courses. In 2019 IEEE Symposium Series on Computational Intelligence (SSCI), Xiamen, China, pp. 442-447. https://doi.org/10.1109/SSCI44817.2019.9002713

[29] Sharma, P., Joshi, S., Gautam, S., Maharjan, S., Khanal, S.R., Reis, M.C., Barroso, J., de Jesus Filipe, V.M. (2022). Student engagement detection using emotion analysis, eye tracking and head movement with machine learning. In International Conference on Technology and Innovation in Learning, Teaching and Education, pp. 52-68. https://doi.org/10.1007/978-3-031-22918-3_5

[30] Zhang, X., Han, L., Zhu, W., Sun, L., Zhang, D. (2022). An explainable 3D residual self-attention deep neural network for joint atrophy localization and Alzheimer's disease diagnosis using structural MRI. IEEE Journal of Biomedical and Health Informatics, 26(11): 5289-5297. https://doi.org/10.1109/JBHI.2021.3066832

[31] Wang, L., Wang, C., Sun, Z., Cheng, S., Guo, L. (2020). Class balanced loss for image classification. IEEE Access, 8: 81142-81153. https://doi.org/10.1109/ACCESS.2020.2991237

[32] Savchenko, A.V., Savchenko, L.V., Makarov, I. (2022). Classifying emotions and engagement in online learning based on a single facial expression recognition neural network. IEEE Transactions on Affective Computing, 13(4): 2132-2143. https://doi.org/10.1109/TAFFC.2022.3188390

[33] Ryumina, E., Dresvyanskiy, D., Karpov, A. (2022). In search of a robust facial expressions recognition model: A large-scale visual cross-corpus study. Neurocomputing, 514: 435-450. https://doi.org/10.1016/j.neucom.2022.10.013

[34] Antoniadis, P., Filntisis, P.P., Maragos, P. (2021). Exploiting emotional dependencies with graph convolutional networks for facial expression recognition. In 2021 16th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2021), Jodhpur, India, pp. 1-8. https://doi.org/10.1109/FG52635.2021.9667014

[35] Gupta, S., Kumar, P., Tekchandani, R.K. (2023). Facial emotion recognition based real-time learner engagement detection system in online learning context using deep learning models. Multimedia Tools and Applications, 82(8): 11365-11394. https://doi.org/10.1007/s11042-022-13558-9

[36] Fakhar, S., Baber, J., Bazai, S., Marjan, S., Jasinski, M., Choudhry, M., Hussain, S. (2022). Smart classroom monitoring using novel real-time facial expression recognition system. Applied Sciences, 12(23): 12134. https://doi.org/10.3390/app122312134

APPENDIX

Terms	Abbreviation	Terms	Abbreviation
DNN	Deep Neural Network	RF	Random Forest
CNN	Convolutional Neural Network	OULAD	Open University Learning Analytics Dataset
FER	Facial Expression Recognition	KNN	K-Nearest Neighbor
LSTM	Long-short term memory	NB	Naïve Bayes
DL	Deep-Learning	CART	Classification and Regression Tree
ML	Machine-Learning	VGG	Visual Geometry Group
ER	Engagement Recognition	Conv2D	2D Convolution Layer
TP	True Positives	FP	False Positives
TN	True Negatives	FN	False Negatives
ROC	Receiver Operating Characteristic	TPR	True Positive Rate
FPR	False Positive Rate	AUC-ROC	Area Under the ROC Curve
FE	Facial Expressions	ER	Engagement Recognition
DT	Decision Tree	MSE	Mean Squared Error
FER-2013	Facial Expression Recognition Challenge 2013		