Journal of Biological Regulators and Homeostatic Agents 2026, 40(3), 124.
https://doi.org/10.65746/jbrha124

Article

Personalized programming support in higher education via emotion-guided prompt generation and GenAI

A. V. Geetha[1,*], T. Mala[2]

[1] Department of Computer Science and Engineering (Artificial Intelligence and Data Analytics), Sri Ramachandra Institute of Higher Education and Research, Porur, Chennai, Tamil Nadu 600116, India
[2] Department of Information Science and Technology, College of Engineering Guindy, Anna University, Chennai, Tamil Nadu 600025, India
*** Corresponding author:** A. V. Geetha, geethaav@sret.edu.in

CITATION

Geetha AV, Mala T. Personalized programming support in higher education via emotion-guided prompt generation and GenAI. Journal of Biological Regulators and Homeostatic Agents. 2026; 40(3): 124.
https://doi.org/10.65746/jbrha124

ARTICLE INFO

Received: 30 May 2026
Revised: 15 June 2026
Accepted: 6 July 2026
Available online: 30 July 2026

Abstract: Personalized programming supports learners by incorporating their emotional states into an Artificial Intelligence (AI) system. However, existing systems face challenges, including a scarcity of emotion-adaptive tutors, limited generative AI tools for recognizing emotions, and a lack of prompt engineering techniques. This study proposes a personalized emotion-aware AI programming assistant that detects the user's facial emotions and generates code explanations using generative AI. We develop a hybrid facial emotion recognition model using ResNet-50 to capture spatial features and LSTM to capture temporal patterns. The model classifies the learner's emotions, which then informs emotion-guided prompt engineering to adjust programming queries based on detected emotions. The modified prompt is processed by a generative language model, displayed to the learner in a supportive and adaptive format. The model is tested in real-time with students. Experimental results show that the emotion-aware AI improved both learning performance and motivation in real-time classroom use, confirming its effectiveness in fostering engagement and better outcomes compared to a normal AI tutor.

Keywords: emotion recognition; generative AI; intelligent tutoring systems; personalized programming support; prompt generation; ResNet-50

1. Introduction

Emotion is a fundamental human physiological response that drives individuals to react to their circumstances. Recent advancements in AI have enabled it to recognize human emotions effectively. Various applications exist across multiple fields, including digital transformation, cloud computing, and intelligent systems, among others [1]. Emotional Intelligence (EI) is crucial for AI and has been studied within the field of Natural Language Processing (NLP) [2]. The rapid progress in AI focuses on integrating EI into these systems. This process involves understanding feelings and responses through numerous modalities, such as facial expressions, speech patterns, and physiological signals [3,4]. The student's emotional state plays a vital role in the education system, and innovative real-time emotion detection and personalized learning systems are being developed to understand and reply to users, providing more intuitive, personalized interactions and engagement [5]. In computer programming, learners often experience a range of emotions in academic environments, which can impact various aspects of their learning performance and cognition [6]. Due to this, an AI-based model for emotional prompting has been developed to generate polite prompts based on the learner's behavior. However, such systems misclassify and produce different outcomes [7]. In recent years, Human-Computer Interaction (HCI) has undergone rapid advancement due to substantial technological developments [8].

Among these developments, AI-driven systems, such as Intelligent Tutoring Systems (ITSs) and Learning Management Systems (LMSs), analyze the student's facial expression and voice emotion to improve the user experience [9,10]. There are several robust emotion recognition models, and experimenters often employ machine learning methods such as multimodal fusion and Deep Learning (DL). As a result, notable approaches include ensemble learning, Recurrent Neural Network (RNN), and Convolutional Neural Network (CNN) for detecting and grading emotions [11]. In the realm of programming education, learners frequently struggle with emotions such as frustration, confusion, and disengagement. These feelings can affect how well they learn and apply new knowledge. Traditional tutoring systems and AI assistants excel at providing technical assistance, but they often overlook the emotional aspect of learning. These systems offer generic responses that fail to identify and adapt to a learner's emotional state [12]. To bridge this gap, this study aims to develop an emotion-aware, personalized AI-based programming assistant that provides an efficient AI assistant to programming learners. This research utilizes ResNet-50 with LSTM to detect and classify users' emotional states from facial images and design a rule-based prompt engineering system that adapts to a program query. The system provides custom, empathetic responses, learning engagement, readability, and emotional support by incorporating emotion-driven prompts into generative AI models. This affectively intelligent assistant strives to build a more responsive, supportive, and human-facilitated learning experience in computer programming.

1.1. Research contributions

- This study develops an emotion-aware AI-based system that detects the user's facial emotion and dynamically adapts code and provides explanations using a generative AI model.
- We propose a hybrid emotion recognition model that integrates ResNet-50 to extract spatial features and LSTM for temporal patterns extraction, resulting in accurate facial emotion classification. This architecture significantly enhances the accuracy and robustness of emotion classification across five emotional states (happy, sad, angry, surprised, neutral).
- The study develops a rule-based prompt modification framework to generate a significant prompt based on the identified emotion of the learners. This framework provides several prompts related to the learner's emotions to enhance learner receptiveness.
- We integrate the emotion-modified prompts with a generative language model, GPT-2, to generate personalized programming explanations. These outputs are technically accurate and emotionally resonant, simulating human-like tutoring responses.

This paper is organized as follows: Section 2 presents a comprehensive literature review on emotion recognition, prompt tuning, and generative AI in educational systems. Section 3 details the proposed methodology and the emotion-aware programming assistant framework. Section 4 discusses the experimental results, including model performance, emotional response adaptation, and comparisons with existing systems. Finally, Section 5 concludes with future research directions and key findings.

2. Literature review

The issue of identifying emotions across multiple modalities was addressed by Khediri et al. [13]. Nevertheless, the identification of student emotions from single-modal data, like facial expressions, was insufficient. In the "Multimodal Intelligent Tutoring Emotion Recognition System" (MITERS), three modalities, such as text, speech, and face, are concurrently integrated in an intelligent affective tutoring system. The system is based on real-time detection of student emotions and provides sufficient feedback. Several DL techniques are employed, including Bidirectional Long Short Memory (BiLSTM) for textual emotion prediction, CNN for speech modality emotion detection, and Deep Convolution Neural Network (DCNN) for face modality emotion detection. Boughida et al. [14] introduced e-learning environments, and most adaptive systems did not mind the emotional state of learners when suggesting actions, obstacles, challenges, or demotivation in learning. During educational activities, the learner's emotions are reflected in the facial expressions of the instructor. For this purpose, an emotion-based probability quantification algorithm was suggested. An adaptation approach was employed, utilizing a set of adopted criteria that vary in weight from one support resource to another, as described in the recommended support resources.

Maddu et al. [15] examined a hybrid prediction model that combines a CNN and the Improved Deep Belief Network (IDBN) to predict online learners' interest based on facial emotions. Furthermore, the model develops adaptive models that account for individual differences in expressing emotions, which could improve accuracy and fairness. Ul Huda et al. [16] investigated how feeding the Large Language Model (LLMs) conflicting emotional cues using various prompting strategies affected the LLM. This study employed no automated model and is restricted to manual prompt feeding. The assessment is based on given instructions as opposed to emotional cues. Every prompting method is examined by evoking a range of emotions popular on ChatGPT 3.5, Gemini, and LLMs. Human-Computer Interaction (HCI) is a realistic cue that is supplied to exert a cognitive burden, rather than relying on an automation process.

Xu et al. [17] examined how different teacher feedback interventions affected the emotional and cognitive interactions of students participating in online group discussions. This study collected online collaborative learners' collaborative texts. Text data from collaborative discussions was automatically coded using the naive Bayes method. The bivariate experiment was created based on the teacher's feedback, with or without emotional content. Emotional input from teachers encouraged the students in both cognitive and emotional engagement, which facilitates the development of cognitive interaction. The suggested group appeared on the multi-branch voice prompt feedback, having a greater impact on emotional and cognitive learners. Garg et al. [18] investigated the prompt teaching structure that affected learning students, such as programming and data analysis. Techniques such as chain prompting, CLEAR framework, and few-shot prompting were covered in the prompt training session. Python data analysis tasks were used to evaluate participants' performance in pre- and post-tests, as well as their programming abilities across three levels of Bloom's taxonomy. An ANOVA revealed significant differences between the groups on post-test scores. Boitel et al. [19] proposed a multi-modal framework called MIST, which combines the modalities of Motion, Image, Speech, and Text from various data sources. The author utilized various deep learning models, including Semi-CNN, ResNet-50, DeBERTa, and 3D-CNN, to analyze the modalities and validate the emotions. Unfortunately, the paradigm struggles to classify the basic emotion. Pourmirzaei et al. [20] designed an affective tutoring system named ATTENDEE (AffecTive Tutoring system based on facial EmotioN recognition and heaD posE Estimation) to customize the learning environment by utilizing facial emotion recognition and estimating learners' head position. This study has also developed the Intelligent Analyzer (IA) device to identify learners' head angles and facial expressions.

Kohnke et al. [21] found that Generative Artificial Intelligence (GenAI) often reduces stress and anxiety, and enhances student motivation, fostering an emotionally supportive learning environment. Nonetheless, some challenges were identified related to both technical and cultural settings. This work highlighted the important role of instruction and experiences provided by the GenAI among students, also emphasizing the need for professional development and ongoing support. It also showed the significance of balance, technology infrastructure, and cultural sensitivity. This study utilized GenAI, which was valuable for maintaining the components of human instruction. It also enhanced the growing body of information on integrating AI into language learning. Shi et al. [22] integrated innovative AI technologies, such as adaptive learning systems and emotion recognition, to improve 274 English Foreign Language (EFL) learners' second language acquisition. By using a pre-post-trial for a randomized control, the study compared traditional methods with AI-enhanced interventions on language competency and emotional self-regulation. These findings showed that the learner significantly utilized AI tools to enhance their language retention and emotional regulation. This study integrated AI technology into learning language programs that can improve linguistic abilities and educational outcomes for customizing learning experiences and emotional involvement.

The above-mentioned existing research focuses on intelligent tutoring systems in programming education that utilize static or traditional AI models, often lacking emotional adaptability. This system in an emotional state of learners usually fails to respond or recognize generic feedback mechanisms that do not address confusion, frustration, or disengagement. While some approaches integrate basic emotion recognition techniques, they often depend on single-modal data or manual prompt engineering, which limits their scalability and personalization capabilities. Furthermore, the inability of current models to dynamically adjust instructional content based on real-time emotional input reduces their effectiveness in supporting learner comprehension and motivation. Existing models typically use unmodified generative AI outputs or isolated emotion detection that fails to exploit the potential of guided emotion learning. The analysis of the existing work is provided in **Table 1**. Thus, a result of the current solution is that it fails the level of empathetic, adaptive support needed for effective programming education. The key challenge is to create an emotionally intelligent, responsive AI tutor that can seamlessly combine facial emotion recognition and generative AI via rule-based prompt tuning. This study aims to tackle these shortcomings by developing a robust, tailored, and generalizable approach for emotionally adaptive programming support, aiming to enhance learner engagement, comprehension, and long-term academic performance.

Table 1. Analysis of existing work in personalized learning based on emotion.

References	Models	Objectives	Modality used	Limitation
Khediri et al. [13]	MITERS	Detects students' emotions during e-learning.	Face, text, speech	Lacks a generative response mechanism.
Boughida et al. [14]	probability-based emotion quantification algorithm	Recognizes learner emotions through facial expressions during educational activities.	Facial Expression	Focused only on facial emotion recognition and didn't generate the prompt.
Maddu et al. [15]	Improved DBN and CNN	The study proposed a model for predicting student engagement among online learners based on facial emotions.	Facial Expression	Lack of adaptive models that differentiate the expressed emotions.
Ul Huda et al. [16]	Large Language Models such as ChatGPT-3.5, Gemini, and other LLMs	This study explored how conflicting emotional cues influence the behavior and output of LLM using different prompting strategies.	Text	Limited to manual prompting without automated emotion detection and lacks integration with real-time adaptive learning.
Xu et al. [17]	Naïve Bayes	Explored the impact of various teacher feedback strategies on students' emotional and cognitive engagement during online group discussions.	Text	Focused solely on textual interaction; lacks emotional or multimodal adaptation.
Garg et al. [18]	Generative AI	The study investigated how structured prompt training using generative AI improves students' data analysis and programming performance in Python.	Text-based prompts entered by students	Does not explore the emotional engagement of the student.
Boitel et al. [19]	DeBERTa, Semi-CNN, ResNet-50, 3D-CNN	The study introduced the multimodal method named MIST to recognize emotions.	Text, image, speech, and text	This study struggled to detect the subtle forms of emotions (fear or surprise).
Pourmirzaei et al. [20]	ATTENDEE (AffecTive Tutoring system based on facial EmotioN recognition and head pose Estimation)	Detecting learners' emotional and attentional states to personalize the e-learning environment.	Face expression and head pose	Emotion recognition based on facial expressions can be highly individual and may misinterpret subtle specific cues.
Kohnke et al. [21]	Generative AI model	This study investigated the students' emotional engagement through generative AI on the emotional engagement.	-	This study utilized effective generative AI but inefficient teaching and learning prompt generation, based on emotions.
Shi et al. [22]	No specified model used	To integrate emotion detection and adaptive learning systems to enhance emotional self-regulation and academic performance in second language learning	Facial expressions (via webcam images)	Focused only on facial emotion data without integrating multi-modal emotional signals

3. Proposed methodology

This study introduces an emotion-adaptive programming assistant that combines facial recognition with generative AI to deliver personalized learning support. This process begins with data collection, where a range of emotion labels, such as anger, happiness, sadness, fear, and surprise, are used in a facial emotion image dataset. These images undergo preprocessing steps, which include normalization using the min-max normalization technique to improve generalization and robustness. Then, the preprocessed images are passed through a developed hybrid facial emotion recognition model, which combines ResNet-50 and LSTM, to detect and classify the emotional state of learners. After that, the learner enters a programming-related query to the proposed Emotion-Guided Prompt Engineering system for emotion-based prompt generation. Finally, the generative AI model GPT-2 generates the code explanations based on the predicted emotion. The overall framework of the proposed study is shown in **Figure 1**.

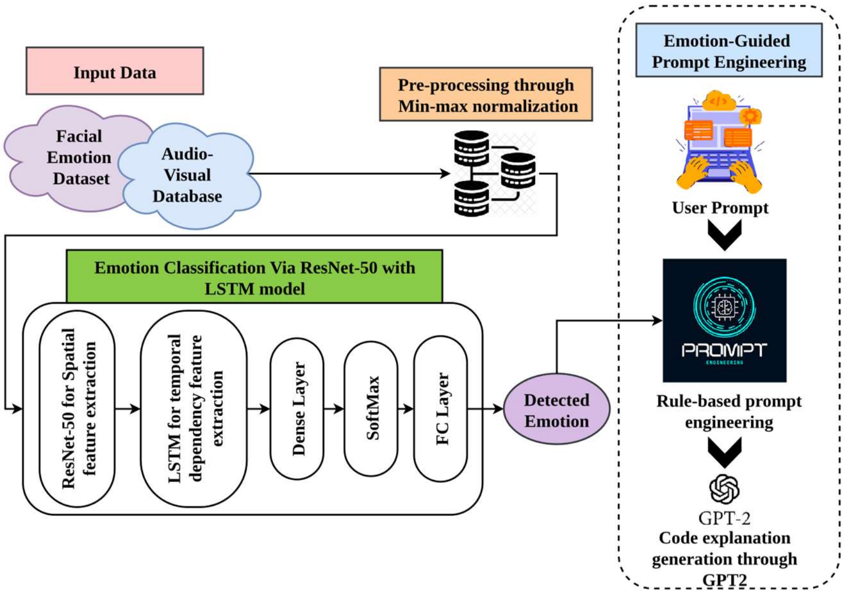


Figure 1. Overall framework of proposed study.

3.1. Dataset description

This study utilizes Audio-Visual Database (RAVDESS) [23], a publicly available dataset that provides a diverse collection of labeled facial expressions across five primary emotion categories: Angry, happy, sad, neutral, and surprised. This dataset comprises 7356 files featuring 24 professional actors (12 female, 12 male), each vocalizing two lexically matched statements in a neutral North American accent. The speech data encompasses expressions of calm, happy, sad, angry, fearful, surprised, and disgusted emotions, while the singing includes expressions of calm, happy, sad, angry, and fearful emotions.

3.2. Preprocessing through min-max normalization

This study utilizes the min-max normalization [24] to ensure that all facial images have a consistent intensity scale. This process standardizes the input data to reduce variability caused by lighting and contrast differences by transforming the pixel values to a uniform scale. Min-Max is a common normalization method that scales the intensity range of each pixel to between 0 and 1.

This is achieved by using the following Equation (1):

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (1)$$

The original pixel value in the image is represented by X, the maximum and minimum pixel intensity values in the image are indicated by X_{min} and X_{max}, respectively, and the normalized pixel value that results is called X_{norm}. To remove intensity discrepancies and improve the model's ability to process inputs, this transformation rescales all pixel values to lie between 0 and 1.

For implementation, all facial images extracted from the RAVDESS dataset were first resized to 224 × 224 pixels to match the input requirements of the pretrained ResNet-50 network. The RGB pixel intensities, originally represented in the range of 0–255, were converted to 32-bit floating-point values before applying min-max normalization. Each pixel value was then divided by 255, resulting in normalized intensity values within the range [0, 1]. After normalization, the facial images were organized into temporal sequences of five consecutive frames, corresponding to the sequence length used by the proposed ResNet-50–LSTM model. This preprocessing pipeline ensures consistent feature extraction across all input samples, accelerates model convergence during training, and enhances the reproducibility of the proposed emotion recognition framework

3.3. Classification of learner's facial emotion through the proposed hybrid facial emotion recognition model

In our work, we combined ResNet-50 with LSTM to effectively classify the facial emotions of programming learners. Here, the backbone network contains ResNet-50 to extract the spatial features (i.e., facial structure, shape, and texture), while the Bidirectional LSTM captures the temporal evolution of facial expressions across consecutive video frames. The input video is first divided into sequences of five consecutive facial frames, where each frame is processed independently by ResNet-50. After feature extraction, the resulting feature vectors are arranged in chronological order to form a temporal sequence, which is then provided as the input to the Bidirectional LSTM. Finally, the classification head, consisting of fully connected dense layers followed by a SoftMax classifier, predicts the learner's facial emotion.

3.3.1. Spatial feature extraction using ResNet-50

The ResNet-50 architecture consists of several convolutional blocks with average pooling, an output layer with a SoftMax layer for classification. It features five main convolutional layers, starting with an initial layer that has 64 filters of size 7, followed by max pooling with a stride of 2.

This is followed by four residual blocks, ending with a fully connected layer for classification. In our study, ResNet-50 proves to be more effective in extracting deep visual features from facial expressions, addressing common vanishing gradient issues in deep networks. The input images progress through multiple convolution and identity blocks, enabling the network to capture discriminative emotional characteristics, including facial textures, contours, and subtle expression variations.

Each frame from the learner's video input is first processed using a pre-trained ResNet-50 model, noted for its strong capability in extracting hierarchical spatial features, such as facial contours, edges, and micro-expressions. The ResNet-50 model's residual connections facilitate effective feature propagation, circumventing vanishing gradient issues in deeper networks. The residual learning function transforms the input into the desired output, enabling the network to bypass the need to learn the entire mapping from scratch. The output of the residual block is represented in Equation (2).

$$Y = F(X, \{W_m\} + X) \tag{2}$$

where X stands for the input, Y is the output of the residual block, F represent residual function, W_m is the weight of the convolutional layer inside the block. After the final global average pooling layer of ResNet-50, each facial frame is represented by a 2048-dimensional feature vector. Since emotions evolve continuously over time, five consecutive facial frames (sequence length = 5) are extracted from each video segment. Each of these five frames is independently passed through ResNet-50, producing five corresponding 2048-dimensional feature vectors. These vectors are then arranged according to their temporal order to construct a feature sequence of size 5 × 2048, where each row corresponds to one video frame and each column represents one extracted spatial feature. During model training, these sequences are processed in batches with an input tensor of dimension (16 × 5 × 2048). This temporal feature sequence preserves both the spatial representation extracted from each frame and the chronological relationship among consecutive frames, providing an appropriate sequential input for temporal modeling by the Bidirectional LSTM.

3.3.2. Temporal dependency feature extraction Via LSTM

In this section, a Bidirectional Long Short-Term Memory (BiLSTM), an extension of the recurrent neural network (RNN), is integrated after the ResNet-50 feature extraction stage to model temporal dependencies among consecutive facial expressions. The BiLSTM receives the ordered sequence of five 2048-dimensional feature vectors generated by ResNet-50, where each feature vector corresponds to one time step in the sequence. By processing the sequence in both forward and backward directions, the BiLSTM captures both past and future contextual information, enabling the model to learn the temporal progression of facial expressions rather than treating each frame independently.

LSTM is particularly well-suited for sequential data due to its gated architecture (forget gate, update gate, cell state, and output gate), which enables the network to retain relevant information while discarding redundant information over extended time intervals. The process of LSTM is defined in Equations (3–7).

$$\mathbb{i}^{(\mathrm{t})} = \sigma\left(\mathfrak{w}_{\mathfrak{x}\mathbb{i}}\mathbb{x}^{(\mathrm{t})} + \mathfrak{w}_{\mathfrak{h}\mathbb{i}}\mathbb{h}^{(\mathrm{t}-1)} + \mathfrak{w}_{\mathfrak{c}\mathbb{i}}\mathbb{c}^{(\mathrm{t})} + b_{\mathbb{i}}\right) \tag{3}$$

$$\mathbb{f}^{(\mathrm{t})} = \sigma\left(\mathfrak{w}_{\mathfrak{x}\mathbb{f}}\mathbb{x}^{(\mathrm{t})} + \mathfrak{w}_{\mathfrak{h}\mathbb{f}}\mathbb{h}^{(\mathrm{t}-1)} + \mathfrak{w}_{\mathfrak{c}\mathbb{f}}\mathbb{c}^{(\mathrm{t})} + b_{\mathbb{f}}\right) \tag{4}$$

$$\mathbb{c}^{(\mathrm{t})} = \mathbb{f}^{(\mathrm{t})}\mathbb{c}^{(\mathrm{t}-1)} + \mathbb{i}, tanh(\mathfrak{w}_{\mathfrak{x}\mathbb{c}}\mathbb{x}^{(\mathrm{t})} + \mathfrak{w}_{\mathfrak{h}\mathbb{c}}\mathbb{h}^{(\mathrm{t}-1)} + b_{\mathbb{c}}) \tag{5}$$

$$\mathbb{o}^{(t)} = \sigma\left(\mathfrak{w}_{\mathfrak{x}o}\mathbb{x}^{(t)} + \mathfrak{w}_{\mathfrak{h}o}\mathbb{x}^{(t-1)} + \mathfrak{w}_{\mathfrak{c}o}\mathbb{x}^{(t)} + b_o\right) \quad (6)$$

$$\mathbb{h}^{(t)} = \mathbb{o}^{(t)}\, tanh(\,\mathbb{c}^{(t)}) \quad (7)$$

where, $\mathbb{x}^{(t)}$ and $\mathbb{h}^{(t)}$ represent the input and hidden vectors with the subscript t indicating the time step, while $\mathbb{i}^{(t)}$ signifies the activation function of the input gate, $\mathbb{f}^{(t)}$ signifies the activation function of the forget gate, $\mathbb{c}^{(t)}$ signifies the activation function of the memory cell and $\mathbb{o}^{(t)}$ denote the activation functions of the output gate. $\mathfrak{w}$ and b signify the weight and bias matrices. Lastly, the hidden states generated by the Bidirectional LSTM summarize the temporal dynamics of the learner's facial expressions across the five-frame sequence. These learned temporal features are subsequently passed to fully connected dense layers, followed by a SoftMax activation function, which classifies the learner's emotional state into one of the five categories: angry, happy, sad, neutral, and surprised.

3.4. Emotion-guided prompt engineering

In modern intelligent tutoring systems, understanding and adapting are essential for the emotional state of learners for developing engagement, motivation, and personalized support. A crucial advancement is emotion-aware prompt tuning, which enables an AI system to adjust its instructional strategies based on learners' emotional feedback. **Figure 2** shows the stepwise architecture of the proposed emotion-based prompt AI system. This process ensures that the generative AI not only delivers accurate content but also provides a supportive learning experience, is more humanized, and supports the learner's emotional state.

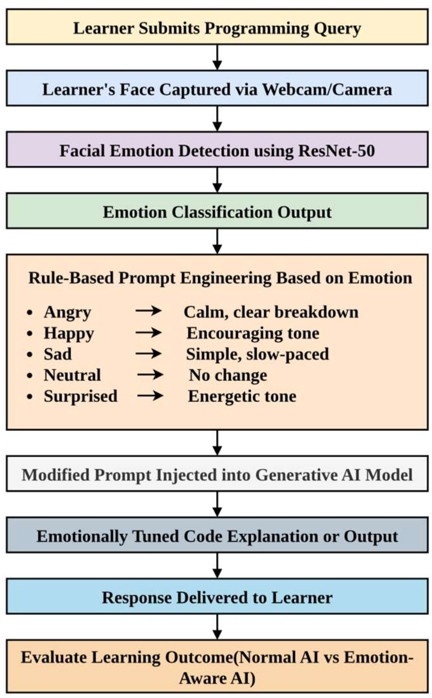


Figure 2. Stepwise architecture of proposed prompt tuning in ai-based learning systems.

3.4.1. Rule-based prompt modification

This section explained the process of prompt generation based on the identified emotions of the learners using the hybrid classification model. First, the learner submits a programming-related query to the system. Second, based on their facial emotion, the proposed AI model generated a code. There are several rules suggested for each emotion. For example, when a learner is detected to be angry, the system modifies the query to say “Let’s break this down clearly and calmly.” This subject shows empathy and offers more structure on a gentle approach. And for a happy learner, the system includes a positive, encouraging tone of something like “Let’s explore this exciting topic together!”. In contrast, if the learner is feeling sad, the query is to be more reassuring, and a soft tone like “We’ll take it slowly with a simple explanation” offers respect for the learner’s emotional state, which makes the content more digestible. If the learner is neutral, the prompt does not modify, acknowledging that no adjustment is needed to the emotion. However, if the learner is surprised, the system encourages exploration like “This sounds exciting! Let’s dive into it!” These emotionally adaptive prompts serve as the input of the next phase for effective code generation using generative AI.

Table 2 presents representative examples of the proposed rule-based emotion-guided prompt generation strategy. Although the learner’s programming query remains unchanged, the prompt is adaptively modified according to the detected emotional state to guide the generative AI in producing responses with an appropriate instructional tone, level of emotional support, and pedagogical style. These examples illustrate how emotion-aware prompt engineering personalizes the interaction while preserving the technical content of the learner’s request.

Table 2. Representative examples of emotion-guided prompt generation.

Detected emotion	Learner query	Emotion-guided prompt	Expected AI response style
Angry	Explain Java loops.	The learner appears frustrated. Please explain Java loops calmly, using simple steps and reassuring language with an easy example.	Calm, structured, reassuring
Happy	Explain Python Boolean values.	The learner is enthusiastic. Explain Python Boolean values using an engaging tone and provide interesting examples to encourage further learning.	Positive, encouraging
Sad	Explain recursion in C.	The learner seems discouraged. Explain recursion slowly using simple language and provide an easy example with step-by-step guidance.	Gentle, supportive
Neutral	Explain arrays in Java.	Explain Java arrays with a standard technical explanation and practical example.	Standard instructional
Surprised	Explain JavaScript operators.	The learner is curious and surprised. Explain JavaScript operators using an energetic tone and include interesting real-world examples.	Energetic, exploratory

3.4.2. Code Generation via GPT-2

At this stage, the emotion-modified prompt is input into a generative language model GPT-2 which is the backbone of the code generation module. Although newer models such as GPT-3 or GPT-4 provide better performance but they need API and cannot run locally. For this reason, we selected GPT-2 for our study due to its open-source availability, ease of local deployment, and reduced computational cost. These characteristics made it a practical choice for developing a proof-of-concept system that can be reproduced, fine-tuned, and deployed without commercial APIs. Additionally, GPT-2 can provide necessary language generation for the task at hand, especially where prompts are well-designed and conditioned on the emotion indicators. This ensures that the system's output is not only accurate but also resonates with the emotion of the learner. By including the emotion sensitivity to the input, the AI response not only corrects the technical solution of the program query but also sends an empathetic and compassionate message to the emotional state of learners. Response delivery is the final stage that focuses on the format of the output, which is presented to maximize emotional engagement and educational impact. The response for an emotion-tailored range varies from an encouraging message to excite the learner to a more supportive, simplified explanation for a sad or frustrated student. This framework equips learners to engage more effectively in complex programming tasks while feeling supported and motivated, regardless of their emotional state.

4. Results and discussion

The classification of facial emotion through the DL model was conducted on a Windows 10 system. The model was developed using Python 3.12 and TensorFlow version 2.17 as the DL framework. The hardware specifications include an Intel Core i5-6500 CPU clocked at 3.20 GHz, 16.0 GB of DDR3 RAM, and Intel(R) HD Graphics 530 for graphics. The model achieved strong performance using a proposed emotion classification model, with key hyperparameters summarized in **Table 3**. The model was trained for 30 epochs with a batch size of 16, which is suitable for limited computational resources and helps introduce gradient noise for better generalization. A learning rate of 0.001 ensures stable fine-tuning of the pretrained ResNet-50 model, which is used for extracting visual features from images resized to 224 × 224 pixels. A sequence length of 5 indicates that five consecutive facial frames are processed to capture short-term temporal dependencies for emotion recognition. The architecture includes bidirectional LSTM layers, with the first bidirectional layer configured with 656 and 384 units for forward and backward directions, respectively, and a total of 4,720,640 parameters, enabling rich temporal feature learning. Dropout was disabled (rate = 0) after empirical tuning because it did not improve validation performance under the adopted training configuration.

Two dense layers follow: the first with 32,896 trainable parameters to learn high-level feature representations, and the second dense layer contains 645 trainable parameters and produces the final emotion class predictions. The values reported for the BiLSTM and dense layers denote the number of trainable parameters and hidden-unit configurations of the corresponding network layers rather than training hyperparameters. They are included to provide a complete description of the model architecture and improve experimental reproducibility. The model is optimized using Adam, an adaptive optimizer known for efficient convergence, and uses categorical cross-entropy as the loss function, suitable for multi-class emotion classification tasks.

Table 3. Hyperparameter and network architecture configuration of the proposed emotion classification model.

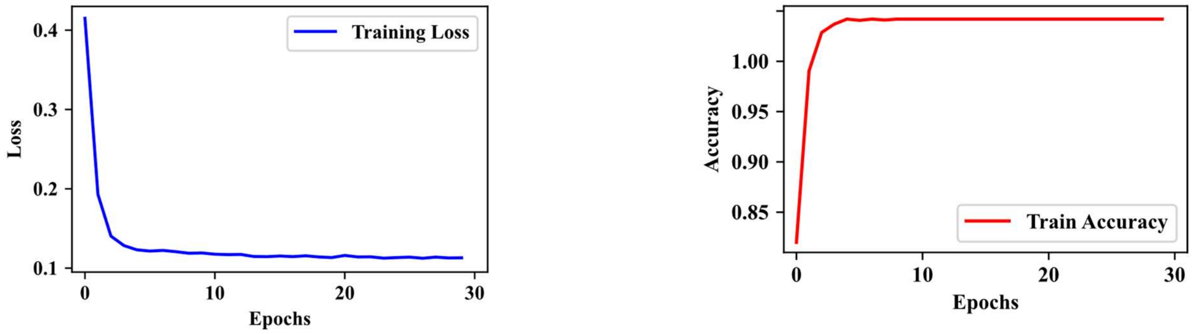
Hyperparameter/network configuration	**Value**
Epochs	30
Batch size	16
Learning rate	0.001
Sequence length	5
Image height, image width	224 × 224
Pretrained model	ResNet-50
BiLSTM trainable parameters	4,720,640
Dropout	0
BiLSTM hidden units (forward, backward)	656, 384
Dense Layer 1 trainable parameters	32,896
Dense Layer 2 trainable parameters	645
optimizer	Adam
loss	Categorical cross-entropy

4.1. Performance evaluation of the proposed emotion classification model

Several metrics were used to evaluate the performance of the proposed emotion classification model. This performance evaluation demonstrates the effectiveness of the proposed classifier model. **Table 4** illustrates the classifier model, ResNet-50 with LSTM, in programming learner facial emotion classification across various metrics, including accuracy, precision, recall, specificity, and F1 score. This evaluation proves the effectiveness of the classification model. The performance of emotion classification throughout 30 epochs of training is shown in **Figure 3**. A steady decrease in training loss indicates that the model is effectively learning from the data over time. The validation loss displays minor fluctuations, maintaining a downward trend that reflects the model's good capability of generalization. During the training process, the final epoch consistently improves the validation accuracy. These trends collectively demonstrate that the model achieves generalization and high learning efficiency, with no significant signs of overfitting.

Table 4. Performance evaluation of the proposed emotion classification model.

Metrics	**Values**
Accuracy	99.63 %
Precision	99.47 %
Recall	99.19 %
Specificity	100 %
F1 score	99.59 %



(a) **(b)**

Figure 3. Performance metrics of the emotion classification model across. **(a)** training loss and **(b)** training accuracy over 30 epochs.

4.1.1. Confusion matrix for emotion classification model

The emotion classification model performs five different classes: surprise, sadness, neutral, happiness, and anger.

Figure 4 shows the confusion matrix. For each class, the proportion of incorrect and correct predictions is displayed in a matrix, where the predicted labels are represented in columns and the true class labels correspond to the rows. The diagonal elements of the matrix (e.g., 258 for Angry, 738 for Happy, 795 for Neutral, 777 for Sad, and 242 for Surprise) indicate the number of instances where the model correctly predicted the emotion.

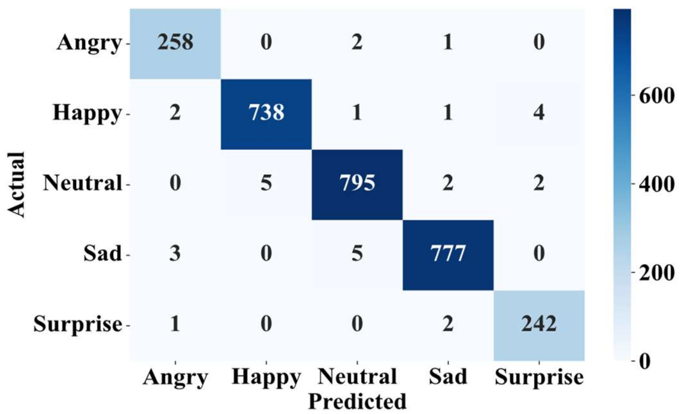


Figure 4. Normalized confusion matrix showing model performance across emotion classes.

For example, the model correctly identified 258 instances of “Angry” emotions and 738 instances of “Happy” emotions. The off-diagonal elements represent misclassifications. or instance, the “Actual” Sad row shows that 3 instances of Sad were misclassified as Angry, and 5 instances were misclassified as Neutral. Similarly, the “Actual” Neutral row shows that 5 instances of Neutral were misclassified as Happy, and 2 as Sad or Surprise. The confusion matrix indicates a strong classification performance with minor confusion among similar emotional expressions.

The observed misclassifications are primarily attributed to the subtle visual similarities between certain emotions, particularly those sharing comparable facial muscle movements and expression intensities.

For example, neutral expressions may occasionally resemble mild happiness or sadness, while surprise and anger can exhibit overlapping facial characteristics under variations in head pose, illumination, facial occlusion, or individual differences in emotional expression. These factors reduce the discriminative information available to the model, resulting in a small number of incorrect predictions.

The confusion between similar emotions may be further reduced by increasing the diversity of the training data, employing attention mechanisms to emphasize discriminative facial regions such as the eyes and mouth, and incorporating complementary contextual cues to improve the model's ability to distinguish subtle emotional variations.

4.1.2. K-fold cross-validation results

Cross-validation is crucial in machine learning for testing the generalizability and power of classification models, where K-fold cross-validation is a commonly utilized technique. It involves splitting the dataset into K, K-sized folds. In terms of evaluating the model, the last fold is used as an evaluation set for a model trained on K-1 folds. This process is repeated K times. Each fold has the chance to be the test set once. We utilized a 10-fold cross-validation is shown in **Table 5**, to ensure robustness and reduce bias. In each round, nine folds were used for training and one for testing, repeated until all folds were tested. The model showed consistently high performance with minimal variation, confirming its reliability and generalization for real-time emotion recognition.

Table 5. Results obtained through K-Fold cross-validation.

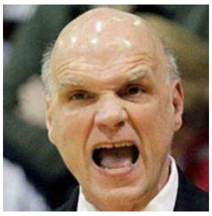
Folds	Accuracy (%)	Precision (%)	Recall (%)	F1 score (%)
Fold 1	99.78	99.31	99.27	99.36
Fold 2	99.72	99.41	99.33	99.75
Fold 3	99.41	99.51	99.23	99.71
Fold 4	99.69	99.46	99.22	99.42
Fold 5	99.70	99.50	99.51	99.65
Fold 6	99.60	99.57	98.87	99.63
Fold 7	99.55	99.29	99.13	99.62
Fold 8	99.63	99.47	99.19	99.59
Fold 9	99.66	99.41	99.09	99.74
Fold 10	99.38	99.53	99.21	99.52

4.2. Analysis of the proposed emotion-aware prompt generation system based on detected facial expressions

Table 6 presents the visual outcomes of the proposed emotion-aware prompt generation system. It demonstrates how the system responds to different facial expressions of learners by generating emotion-sensitive explanations for programming-related queries. It begins with input frames, which are facial images captured from learners during interaction. These images are analyzed by the proposed hybrid deep learning model, combining ResNet-50 with LSTM to detect the learner's emotional state, such as angry, sad, happy, surprised, or neutral. Once the emotion is identified, the learner's programming-related query is processed by a rule-based prompt engineering mechanism. This mechanism modifies the query based on the detected emotion to ensure the prompt is emotionally appropriate; for instance, it adds calming and structured language for angry learners or a cheerful and engaging tone for happy learners. The emotion-adjusted prompt is then fed into a generative language model (GPT-2), which generates a code explanation or instructional response that is not only technically accurate but also emotionally aligned with the learner's current state. The final output is delivered in a supportive, human-like tone, improving learner engagement and emotional comfort. This figure visually demonstrates how the system integrates emotion recognition with generative AI to produce personalized, empathetic responses that enhance the overall learning experience.

Table 6. Outcomes of the proposed Emotion-aware prompt AI system.

Input frames	**Explanation generation via emotion-aware ai system**
	Detected Emotion: "Happy" The user's detected emotion is "Happy". The user's question is: "Can you explain Python Booleans?" "Use cheerful and light-hearted language. Make it enjoyable and encouraging." boolean value is expressed using the boolean keyword, i.e. bool (true) or bool (false). Boolean operations are: is, is not, as, as not, not as, not as not, is_not, is_not is _not: returns ture if the boolean value is false; otherwise returns false. is_not: returns true if the boolean value is false; otherwsie returns false. is_true: retruns true if the boolean value is true; otherwise returns false. is_false: returns true if the boolean value is flase; otherwise retruns false. is_not_true: retruns true if the boolean value is not true; otherwise returns false…

	Detected Emotion: "Angry"
	
	The user's detected emotion is "Angry".
	
	The user's question is: "Can you explain JavaScript Operators with code?"
	
	"Be calm and composed. Avoid complexity and reassure the user.""
	
	Operators in JavaScript are symbols that perform operations on variables and values. They are essential for performing calculations, comparisons, assignments, and logical operations.
	
	function add (a, b) {
	return a + b;
	}
	
	The following JavaScript code shows how to add a string to another string:
	function add Two Strings (str 1, str 2) {
	var str12= str1 + str2;
	var str1 = str1 + " ";
	var str2 = str2 + " ";
	return str12 + str1 +str2;
	}

4.3. Real-time analysis of the proposed emotion-aware prompt generation system

4.3.1. Test subjects

The study was conducted at the Loyola Institute of Technology and Science, India, with 40 first-year undergraduate students from the Department of Computer Science and Engineering (CSE). We divided the students into two groups: Group A (20 students) is suggested to utilize Normal AI, which provides generic programming support without emotional adaptation. Group B (20 students) is suggested to utilize the Emotion-Aware AI system. This detects learners' emotional states and delivers emotionally responsive prompts. **Figure 5** shown the data collection from the engineering college students in real-time. Both groups took a pre-test to measure baseline programming knowledge, followed by a 45-day learning period, and a post-test is validated by the class tutor.

Figure 5. Data collection from the engineering college students in real-time. A group of students was made to learn the curriculum using the developed model.

To evaluate learner motivation, a five-point Likert scale was administered after the completion of the 45-day learning period. Students were asked to rate their overall learning motivation and engagement while interacting with their assigned AI tutoring system. The scale ranged from 1 (Very Low Motivation) to 5 (Very High Motivation), where higher scores indicate greater learning motivation. The motivation score for each experimental group was calculated as the arithmetic mean of the individual student ratings. **Table 7** summarizes the motivation scoring scale used in the study, while **Table 8** presents the comparative analysis of learning performance and motivation scores for the two experimental groups.

Table 7. Motivation scoring scale used in the real-time learning experiment.

Score	Interpretation
1	Very Low Motivation
2	Low Motivation
3	Moderate Motivation
4	High Motivation
5	Very High Motivation

Table 8. Analysis of learning performance and motivation scores of Normal AI vs. Emotion-Aware AI evaluated from the real-time student groups.

Groups	Pre-test learning performance	Post-test learning performance	Motivation scores (1–5)
Group A (Normal AI)	45 %	63 %	3.2
Group B (Emotion-Aware AI)	42 %	74 %	4.4

This highlights clear differences in learning performance and motivation between the two groups. At the pre-test stage, both groups showed comparable baseline knowledge, with Group A (Normal AI) scoring 45 % and Group B (Emotion-Aware AI) scoring 42 %, indicating similar starting levels.

After the intervention, Group A improved moderately to 63 %, achieving an 18-point gain, while Group B showed a more substantial increase, reaching 74 % with a 32-point gain. This indicates that the Emotion-Aware AI system was more effective in enhancing student learning outcomes. Additionally, motivation levels varied significantly among the groups. Group A reported a moderate score of 3.2 on a five-point scale, whereas Group B achieved a much higher score of 4.4. This suggests that our proposed emotion-aware system not only supported stronger academic performance but also fostered greater engagement and enthusiasm among learners.

4.3.2. Real-time interface of the proposed emotion-aware AI system detecting learners' emotions during programming tasks

The real-time interface of our proposed emotion-aware AI system demonstrates how learners' facial expressions are continuously analyzed during programming tasks. As shown in **Figure 6**, the system detects emotions directly from the learner's webcam feed. Each frame undergoes preprocessing before being classified by the ResNet-50 with LSTM model, which ensures both spatial and temporal emotional features are captured. The identified emotion is displayed as an overlaid label along with the confidence score, allowing the system to adapt programming support dynamically.

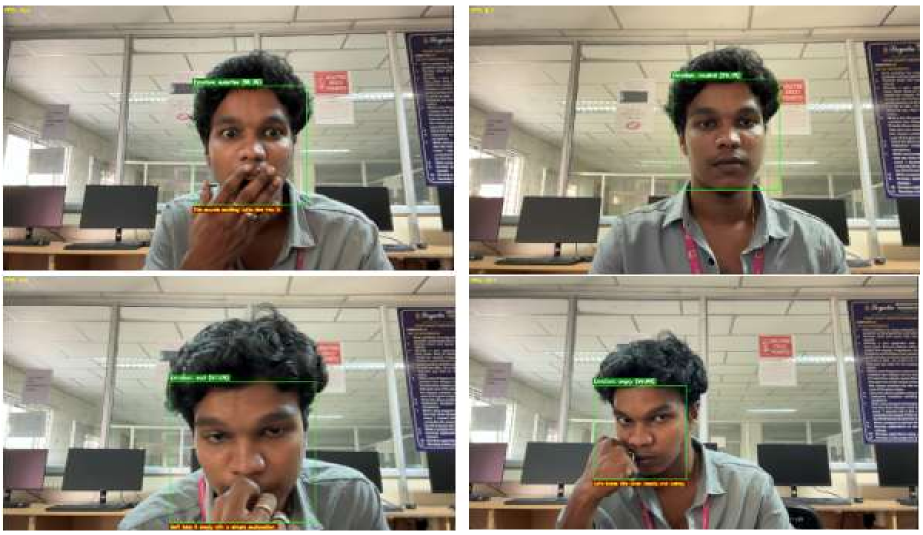


Figure 6. Real-time interface of the proposed emotion-aware AI system detecting and classifying learners' facial emotions during programming tasks for adaptive prompt generation.

For instance, when the learner exhibits confusion or anger, the AI modifies its response to be more calming and structured, whereas for surprise or neutral states, it provides supportive or exploratory guidance. This real-time adaptability highlights the system's capacity to foster emotionally intelligent learning experiences by synchronizing instructional feedback with the learner's affective state.

4.3.3. Comparative analysis of proposed model with existing models

Table 9 presents the comparative analysis of the proposed model with various existing models, including CNN [13], Probabilistic Model [14], IDBN with CNN [15], LLM [16], and Naïve Bayes [17]. These models are evaluated using standard performance metrics, including accuracy, precision, recall, specificity, and F1-score. Among all the compared methods, the proposed ResNet-50 with LSTM model achieves the highest performance, attaining an accuracy of 99.63 %, precision of 99.47 %, recall of 99.19 %, specificity of 100.00 %, and an *F1*-score of 99.59 %. Furthermore, the proposed model demonstrates an accuracy gain ranging from 0.72 % over the LLM model to 5.76 % over the Naïve Bayes model, highlighting its consistent improvement across diverse baseline approaches. The statistical significance analysis further validates these performance gains, with t-values ranging from 4.96 to 8.61 and corresponding p-values ranging from 0.0065 to 0.0003, all of which are below the significance threshold of 0.05. These results indicate that the observed improvements are statistically significant and unlikely to have occurred by chance. Overall, the findings demonstrate that the proposed ResNet-50 with LSTM model provides superior emotion recognition capability, thereby enabling more accurate emotion-aware prompt generation and enhancing personalized, emotionally intelligent support in AI-based programming tutoring systems.

Table 9. Evaluation metrics of proposed and existing models.

References	Models	Accuracy (%)	Precision (%)	Recall (%)	Specificity (%)	*F1*-score (%)	Accuracy gain (%)	*t*-value	*P*-value
Khediri et al. [13]	CNN	96.84	96.12	95.76	97.31	95.94	2.79	6.84	0.0012
Boughida et al. [14]	Probabilistic Model	95.71	95.03	94.82	96.21	94.92	3.92	7.48	0.0007
Maddu et al. [15]	IDBN with CNN	98.36	98.02	97.84	98.71	97.93	1.27	5.92	0.0031
Ul Huda et al. [16]	LLM	98.91	98.67	98.42	99.11	98.54	0.72	4.96	0.0065
Xu et al. [17]	Naïve Bayes	93.87	93.24	92.68	94.51	92.96	5.76	8.61	0.0003
Ours	ResNet-50 with LSTM	99.63	99.47	99.19	100.00	99.59	-	-	-

4.4. Discussion

The proposed Emotion-Aware Prompt Generation System addresses a critical gap in intelligent tutoring systems by introducing emotional adaptability into programming assistance. Traditional AI models often provide generic responses that lack emotional sensitivity, which can negatively impact learner motivation, comprehension, and engagement, particularly in emotionally taxing subjects such as programming. This study overcomes those limitations by integrating a hybrid deep learning model (ResNet-50 + LSTM) for facial emotion recognition and combining it with rule-based prompt engineering and generative AI (GPT-2). Through the integration of spatial and temporal facial features, the proposed model reliably identifies emotional states, including angry, sad, happy, neutral, and surprised. The high classification accuracy (99.63 %) and strong performance across precision, recall, and specificity metrics validate the robustness of the model in real-world learning settings. The system's core innovation lies in emotion-guided prompt tuning, where the detected emotion is used to slightly alter the phrasing of a programming query to match the learner's emotional state. This modification improves emotional alignment between the AI's response and the user's current affective state. For example, angry users receive calm and structured responses, while users experiencing sadness are given gentle and encouraging explanations. This empathy-driven approach enhances the perceived supportiveness of the AI, making learning more inclusive and emotionally intelligent. Furthermore, comparative analysis with standard AI systems shows that emotion-aware responses are consistently more engaging, emotionally relevant, and helpful. In addition, emotion distribution analysis reveals that learners most frequently present neutral, happy, or sad expressions, reinforcing the importance of real-time adaptation to subtle emotional cues. Overall, the study shows the potential of emotionally responsive AI to improve learning.

The selection of GPT-2 provides several practical advantages for the proposed educational system. Its relatively small model size enables efficient local deployment with modest computational resources, reducing inference latency and preserving learner privacy by avoiding cloud-based processing. These characteristics make GPT-2 well suited for classroom environments and educational institutions with limited computational infrastructure. However, compared with larger commercial language models such as GPT-3.5/4, GPT-2 has comparatively weaker contextual understanding, reasoning capability, and programming knowledge, which may affect the quality of explanations for complex programming problems. Nevertheless, GPT-2 is sufficient to demonstrate the effectiveness of the proposed emotion-aware prompt generation framework, while the underlying architecture remains compatible with more advanced language models that could further enhance response quality and pedagogical adaptability.

Despite the strong performance, some limitations remain. First, the study relied on the RAVDESS dataset, which contains acted emotions and may not fully capture the subtle, spontaneous, and context-dependent emotional expressions exhibited by learners in real classroom environments. Consequently, a distribution gap may exist between acted and authentic emotions, potentially affecting the model's generalization in practical educational settings. To reduce this deviation, the model can be fine-tuned using naturalistic classroom emotion datasets with human-annotated labels, and domain adaptation or transfer learning techniques can be employed to better align the learned representations with real-world learner behaviors. In addition, incorporating multimodal cues, such as speech, gaze, or learner interaction patterns, can provide complementary emotional information and improve the robustness of emotion recognition in authentic classroom scenarios. Second, dropout was not applied due to architectural constraints and dataset size, though overfitting was mitigated with transfer learning, careful hyperparameters, and validation monitoring; additional regularization strategies will be explored. Third, the emotion set was limited to five broad categories (angry, happy, sad, neutral, and surprised), excluding learning-specific affective states such as frustration, boredom, confusion, engagement, and curiosity that frequently occur during programming learning.

Future work will expand the emotion recognition module by incorporating educational emotion datasets and human-annotated classroom recordings to recognize these additional affective states. The emotion-aware prompt generation mechanism will also be extended to generate instructional feedback tailored to these richer emotional categories, enabling more personalized and adaptive programming support. Moreover, model outputs were not validated against independent human ratings, which will be added for stronger reliability. Finally, GPT-2 was chosen for feasibility, but more advanced generative models (e.g., GPT-3/4 or open LLMs) could further improve fluency and adaptability. Addressing these issues will enhance robustness, ecological validity, and educational impact in future work.

5. Conclusion

This study introduced a novel emotion-aware AI system that utilized facial emotion recognition, rule-based prompt engineering, and generative language models to deliver personalized, emotionally adaptive responses to programming learners. By combining ResNet-50 for spatial feature extraction with LSTM for temporal analysis, the system accurately classifies learner emotions and adjusts programming queries accordingly. The resulting prompts are processed by GPT-2 to generate emotionally aligned code explanations, offering both technical accuracy and empathic support. The proposed classification model achieved an emotion classification accuracy of 99.63 %, outperforming existing methods, and the real-time validation with undergraduate learners demonstrated clear educational benefits: students using the emotion-aware AI achieved higher post-test performance and stronger motivation than those using a normal AI tutor. These findings highlight that emotionally adaptive AI can transform programming education by making learning more engaging, supportive, and effective.

Author contributions: Conceptualization, AVG and TM; methodology, AVG; software, AVG; validation, AVG ,TM; formal analysis, AVG; investigation, AVG; resources, TM; data curation, TM; writing—original draft preparation, TM; writing—review and editing, AVG; visualization, TM; supervision, TM; project administration, AVG; funding acquisition, TM. All authors have read and agreed to the published version of the manuscript.

Funding: None.

Ethical approval: Not applicable.

Informed consent statement: Not applicable.

Data availability statement: The data and materials in this article are sourced from Audio-visual Database: https://zenodo.org/records/1188976.

Conflict of interest: The authors declare no conflict of interest.

References

1. Gamage G, De Silva D, Mills N, et al. Emotion AWARE: An artificial intelligence framework for adaptable, robust, explainable, and multi-granular emotion analysis. Journal of Big Data. 2024; 11(1): 93. doi: 10.1186/s40537-024-00953-2

2. Gu Z, He K. Affective prompt-tuning-based language model for semantic-based emotional text generation. International Journal on Semantic Web and Information Systems (IJSWIS). 2024; 20(1): 1–19. doi: 10.4018/IJSWIS.339187
3. Lee JP, Jang H, Jang Y, et al. Encoding of multi-modal emotional information via personalized skin-integrated wireless facial interface. Nature Communications. 2024; 15(1): 530. doi: 10.1038/s41467-023-44673-2
4. Lai T, Zeng X, Xu B, et al. The application of artificial intelligence technology in education influences Chinese adolescent's emotional perception. Current Psychology. 2024; 43(6): 5309–5317. doi: 10.1007/s12144-023-04727-6
5. Govea J, Navarro AM, Sánchez-Viteri S, et al. Implementation of deep reinforcement learning models for emotion detection and personalization of learning in hybrid educational environments. Frontiers in Artificial Intelligence. 2024; 7: 1458230. doi: 10.3389/frai.2024.1458230
6. Atman Uslu N, Onan A. Exploring students' emotions towards programming: Analysing sentiments using concurrent conversion mixed methods. Education and Information Technologies. 2025; 30(14): 19879–19909. doi: 10.1007/s10639-025-13482-z
7. Vinay R, Spitale G, Biller-Andorno N, et al. Emotional prompting amplifies disinformation generation in AI large language models. Frontiers in Artificial Intelligence. 2025; 8: 1543603. doi: 10.3389/frai.2025.1543603
8. Xu Y, Lin YS, Zhou X, et al. Utilizing emotion recognition technology to enhance user experience in real-time. Computing and Artificial Intelligence. 2024; 2(1): 1388–1388.
9. Cambria E, Zhang X, Mao R, et al. SenticNet 8: Fusing emotion AI and commonsense AI for interpretable, trustworthy, and explainable affective computing. In: International Conference on Human-Computer Interaction. 2024. pp. 197–216. doi: 10.1007/978-3-031-76827-9_11
10. Zheng J, Li S, Wang T, et al. Unveiling emotion dynamics in problem-solving: a comprehensive analysis with an intelligent tutoring system using facial expressions and electrodermal activities. International Journal of Educational Technology in Higher Education. 2024; 21(1): 33. doi: 10.1186/s41239-024-00462-5
11. Narimisaei J, Naeim M, Imannezhad S, et al. Exploring emotional intelligence in artificial intelligence systems: A comprehensive analysis of emotion recognition and response mechanisms. Annals of Medicine and Surgery. 2024; 86(8): 4657–4663. doi: 10.1097/ms9.0000000000002315
12. Sargazi Moghadam T, Darejeh A, Delaramifar M, et al. Toward an artificial intelligence-based decision framework for developing adaptive e-learning systems to impact learners' emotions. Interactive Learning Environments. 2024; 32(7): 3665–3685. doi: 10.1080/10494820.2023.2188398
13. Khediri N, Ben Ammar M, Kherallah M, et al. A real-time multimodal intelligent tutoring emotion recognition system (MITERS). Multimedia Tools and Applications. 2024; 83(19): 57759–57783. doi: 10.1007/s11042-023-16424-4
14. Boughida A, Kouahla MN, Lafifi Y, et al. emoLearnAdapt: A new approach for an emotion-based adaptation in e-learning environments. Education and Information Technologies. 2024; 29(12): 15269–15323. doi: 10.1007/s10639-023-12429-6
15. Maddu RBR, Murugappan S. Online learners' engagement detection via facial emotion recognition in online learning context using hybrid classification model. Social Network Analysis and Mining. 2024; 14(1): 43. doi: 10.1007/s13278-023-01181-x
16. Ul Huda N, Sahito SF, Gilal AR, et al. Impact of Contradicting Subtle Emotion Cues on Large Language Models with Various Prompting Techniques. International Journal of Advanced Computer Science & Applications. 2024; 15(4). doi: 10.14569/IJACSA.2024.0150442
17. Xu W, Yang LY, Liu X, et al. Examining the effects of different forms of teacher feedback intervention for learners' cognitive and emotional interaction in online collaborative discussion: A visualization method for process mining based on text automatic analysis. Education and Information Technologies. 2024, 29(6): 6525–6551.
18. Garg A, Soodhani KN, Rajendran R, et al. Enhancing data analysis and programming skills through structured prompt training: The impact of generative AI in engineering education. Computers and Education: Artificial Intelligence. 2025; 8: 100380. doi: 10.1016/j.caeai.2025.100380
19. Boitel E, Mohasseb A, Haig E, et al. MIST: Multimodal emotion recognition using DeBERTa for text, Semi-CNN for speech, ResNet-50 for facial, and 3D-CNN for motion analysis. Expert Systems with Applications. 2025; 270: 126236. doi: 10.1016/j.eswa.2024.126236
20. Pourmirzaei M, Montazer GA, Mousavi E, et al. ATTENDEE: An AffecTive Tutoring system based on facial EmotioN recognition and heaD posE Estimation to personalize e-learning environment. Journal of Computers in Education. 2025; 12(1): 65–92. doi: 10.1007/s40692-023-00303-w

21. Kohnke L, Moorhouse BL. Enhancing the emotional aspects of language education through generative artificial intelligence (GenAI): A qualitative investigation. Computers in Human Behavior. 2025; 167: 108600. doi: 10.1016/j.chb.2025.108600
22. Shi L. The integration of advanced AI-enabled emotion detection and adaptive learning systems for improved emotional regulation. Journal of Educational Computing Research. 2025; 63(1): 173–201. doi: 10.1177/07356331241296890
23. Livingstone Steven R, Russo Frank A. The Ryerson audio-visual database of emotional speech and song (RAVDESS). 2018. Available online: https://zenodo.org/records/1188976 (accessed on 12 January 2026).
24. Umar MA, Chen Z, Shuaib K, et al. Effects of feature selection and normalization on network intrusion detection. Data Science and Management. 2025; 8(1): 23–39. doi: 10.1016/j.dsm.2024.08.001