IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 9, September 2025

DOI: 10.17148/IJARCCE.2025.14928

"Enhancing multi model emotion detection using deep learning and machine learning"

Srushti S Rao[1], Dr. K Balaji[2]

PG Student, Department of M.C.A, Surana College (Autonomous), Kengeri Bangalore, India[1]

Assistant Professor, Department of M.C.A, Surana College (Autonomous), Kengeri, Bangalore, India[2]

Abstract: Emotions play a vital role in human communication, influencing decision-making, social interaction, and overall well-being. The ability to automatically recognize emotions across multiple modalities has become increasingly important in fields such as mental health monitoring, intelligent customer support, and affective computing. This work proposes a Multimodal Emotion Recognition System that integrates speech, text, and facial expression analysis to achieve a more reliable and comprehensive understanding of human emotions. For speech, features such as Mel Frequency Cepstral Coefficients (MFCCs) and pitch are extracted and analyzed using Random Forest classifiers along with Deep Neural Networks (DNNs) for improved performance. Text-based emotion recognition leverages the contextual learning capabilities of Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) models to capture linguistic nuances. Facial expression recognition is conducted using Convolutional Neural Networks (CNNs), enhanced with wavelet transforms for better feature representation. The fusion of these modalities helps address the limitations of single-source emotion detection, leading to more accurate and holistic recognition. The proposed system is deployed as a user-friendly web application built with Flask, HTML, and CSS, making it accessible for practical use. This research contributes to the advancement of multimodal affective computing and highlights the potential of integrated ML and DL approaches for real-world emotion-aware applications.

INTRODUCTION

Emotions are fundamental to human experience, shaping how individuals communicate, make decisions, and build relationships. They are expressed through multiple channels, including speech, written text, and facial expressions, each carrying unique and valuable information about a person's emotional state. Because emotions influence almost every aspect of daily life, accurately identifying them has widespread importance across numerous fields such as healthcare, education, customer service, and entertainment. For instance, in healthcare, effective emotion recognition can assist in monitoring mental health, identifying mood disorders, and improving patient care. In educational settings, recognizing students' emotional states helps tailor learning experiences to enhance engagement and motivation. In customer service, understanding emotions enables more personalized and empathetic interactions, improving satisfaction and loyalty. Likewise, in entertainment and gaming, detecting player emotions can make experiences more immersive and responsive. Despite the critical role emotions play, traditional emotion recognition systems often rely on a single modality, such as either speech, text, or facial analysis. This unimodal approach is limited because human emotions are complex, subtle, and rarely expressed through just one channel. For example, a person may verbally express happiness while their tone or facial expression reveals stress or uncertainty. By focusing on a single source, such systems can miss important emotional nuances or produce inaccurate interpretations. This limitation has driven the development of multimodal emotion recognition, which integrates multiple sources of emotional information—typically speech signals, textual content, and facial expressions—to provide a richer, more accurate understanding of affective states.

Multimodal emotion recognition systems work by combining and analysing data from these diverse channels, capturing subtle variations that might be overlooked when considering each modality individually. The integration of multimodal data offers a more holistic representation of emotions, increasing both accuracy and robustness. Recent progress in machine learning, especially deep learning, has enabled these systems to process vast and complex data effectively. Techniques like convolutional neural networks (CNNs), recurrent neural networks (RNNs), and transformer models are used to extract meaningful features from images, audio, and text. Additionally, fusion methods—where inputs from different modalities are combined at either the feature or decision-making level—help enhance emotion detection by leveraging complementary information.

The impact of these technological advances extends beyond improved recognition rates. Embedding emotional intelligence into human-computer interactions allows machines to better understand and respond to human feelings, fostering more natural, empathetic communication. Applications include virtual assistants that detect user frustration or satisfaction, healthcare tools that monitor emotional well-being in real-time, and educational platforms that adapt teaching strategies based on learner emotions. Such affect-aware systems make digital experiences more personalized and socially

DOI: 10.17148/IJARCCE.2025.14928

meaningful, helping bridge the gap between human emotional complexity and artificial intelligence.
As technology becomes increasingly integrated into everyday life, the ability of machines to perceive and interpret emotions will be key to creating more intuitive and effective interactions. Multimodal emotion recognition represents a pivotal step toward this goal by capturing the richness of human emotional expression through multiple channels and employing advanced learning algorithms. This approach enhances the quality of human-computer interfaces and holds great promise for transforming how people and machines communicate, collaborate, and connect in the future.

LITERATURE REVIEW

[1] In this paper, emotion recognition is challenging due to the subtle and varied nature of human expressions. CNNs are effective but complicated by diverse architectures, datasets, and hyperparameters. This study improves performance through dataset preparation, hardware-based model selection, and training strategies like transfer learning and gradual freezing, achieving 92.6% accuracy on diverse emotional states.

[2] In this paper, the authors propose an automatic speech emotion classification algorithm using a Convolutional Neural Network (CNN) model based on LeNet. The study enhances existing preprocessing methods and fine-tunes network models to realize effective recognition of emotional states from speech signals. Experimental results demonstrate that the CNN-driven classification method achieves higher recognition rates for speech emotions. This study provides a valuable reference for improving emotion recognition systems and optimizing human-computer interactions.

[3] In this paper, the authors propose an automatic speech emotion classification algorithm using a Convolutional Neural Network (CNN) model based on LeNet. The study enhances existing preprocessing methods and fine-tunes network models to realize effective recognition of emotional states from speech signals. Experimental results demonstrate that the CNN-driven classification method achieves higher recognition rates for speech emotions. This study provides a valuable reference for improving emotion recognition systems and optimizing human-computer interactions.

[4] In this paper, the authors present a new approach to emotion classification from multiple modalities, such as text, images, and speech, which is complex due to subtle and varied emotional expressions. The study focuses on distinguishing emotions using Convolutional Neural Networks (CNNs) and Region Convolutional Neural Networks (R-CNNs). Motivated by the need for accurate emotion recognition, the proposed method compares the testing time and accuracy of pretrained models, including AlexNet, ResNet-50, and GoogLeNet. The proposed model demonstrates reduced training and testing durations while maintaining high accuracy, achieving 98.50% with CNN and 95.50% with R-CNN in emotion classification experiments.

Comparative Analysis of Related Work

Table 2.1: Comparative Analysis

Sl.No	Author(s)	Algorithms/Techniques	Performance Measures
1.	Tong Zhu; Leida Li; Jufeng Yang;Sicheng Zhao; Xiao Xiao	Multi-Level SemanticReasoning Network	92.6%
2.	Sharmeen M.Saleem Abdullah Abdullah, Siddeeq Y. Ameen Ameen, Mohammed A. M.Sadeeq, Subhi Zeebaree	Multimodal Emotion Recognition using DeepLearning	95%
3.	M. D. Hssayeni and B.Ghoraani	Automated momentaryestimation of positive and negative affects (PA and NA)	78%
4.	P. D. Mahendhiran and S. Kannimuthu	Polarity Classification in Multimodal Sentiment Analysis	96.07%
5.	D. Nguyen et al.	Deep Auto-Encoders With Sequential Learning	80%

Objectives of the Present Study

The objectives of the proposed project are as follows:

- To develop a comprehensive multimodal emotion classification system that accurately identifies human emotions from speech, text, and facial expressions.
- To apply advanced machine learning and deep learning techniques, includingCNNs for image and audio data and BERT for textual data.
- To integrate multiple modalities (speech, text, and facial expressions) to achievehigh accuracy and robustness in

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 9, September 2025

DOI: 10.17148/IJARCCE.2025.14928

emotion recognition.

- To create a web application for real-time emotion recognition.

METHODOLOGY

4.1 Methodology Used

We developed a multimodal emotion recognition system that uses speech, text, and facial expression data. The dataset was collected from open-source resources including emotional speech databases, social media text, conversation transcripts, and facial image datasets. Each type of data was preprocessed before training. For speech, features like MFCCs and pitch were extracted and enhanced with noise reduction. Text data was cleaned, tokenized, and converted into numerical vectors using GloVe embeddings. Images for facial expressions were resized, normalized, and processed using CNNs to detect facial features.

For classification, different models were applied to each modality. Speech-based emotions were recognized using Random Forest and Deep Neural Networks to capture audio patterns. Text-based emotion recognition used LSTM networks and Transformer models like BERT to understand context and emotion in language. Facial expressions were classified using CNNs that learned to detect gestures such as smiles or frowns. The predictions from all three modalities were fused using an ensemble approach, improving overall accuracy.

The system was evaluated using accuracy, precision, recall, F1-score, and confusion matrices. Accuracy measured overall correctness, precision reduced false positives, recall captured all positive cases, and the F1-score balanced precision and recall for imbalanced data. A confusion matrix was used to identify strengths and weaknesses across emotion classes.

Training was done with an 80-20 train-test split, and cross-validation ensured generalization. Hyperparameters were optimized using grid and random search techniques. The system was deployed through a Flask-based web application, with a simple HTML/CSS frontend. Users can enter text, upload voice recordings, or submit images, and the backend processes the input to provide real-time emotion predictions.

SYSTEM DESIGN

Architecture of the proposed system

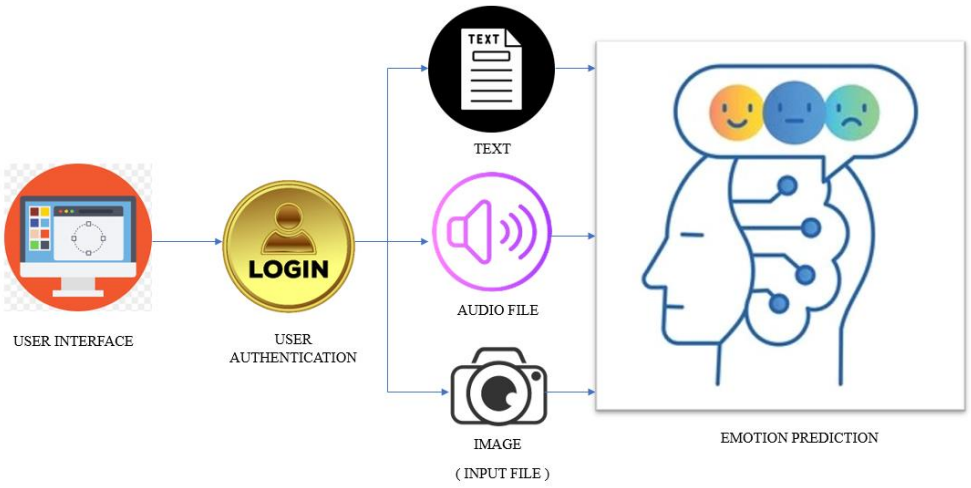


Figure 5.1: Architecture of the Proposed System

1. **User Interface**

- The front-end platform where users interact with the system.
- Provides options to upload input data (text, audio, or image).
- Ensures a seamless and intuitive experience for users.

2. **User Authentication**

- Validates user credentials to ensure secure access.
- Prevents unauthorized use of the system and protects data integrity.

3. **Input Modalities (Text, Audio, Image)**

Accepts input in three forms:

- **Text**: Sentences or paragraphs for text-based emotion analysis.

DOI: 10.17148/IJARCCE.2025.14928

- **Audio**: Voice recordings for speech-based emotion recognition.
- **Image**: Facial expressions captured in images for visual emotion detection.

4. Emotion Prediction Engine

- Integrates machine learning or deep learning models to handle multimodal data.
- Outputs emotion categories like happy, sad, angry, etc., based on the input prov

Deep Learning Algorithms

- **Recurrent Neural Networks (RNNs):**

RNNs are designed for sequence-based data such as speech and text. They can remember past inputs, which helps in detecting emotions over time. For speech, they analyze features like MFCCs to capture tone variations. For text, they follow the order of words to find the emotional meaning in sentences.

- **Long Short-Term Memory Networks (LSTMs):**

LSTMs are an improved form of RNNs that handle long-term context better. They use memory cells and gates to decide what information to keep or forget. In speech, they track pitch, tone, and energy changes over longer recordings. In text, they can follow emotions across full dialogues or long passages.

- **Convolutional Neural Networks (CNNs):**

CNNs identify patterns in images or signals. For facial expressions, they detect features like eyes, lips, or face shapes to recognize emotions. For speech, audio signals can be turned into spectrograms, and CNNs learn patterns to find emotional cues.

Machine Learning Algorithm

- **Random Forest:**

Random Forest is an ensemble method that combines many decision trees. It is reliable for emotion recognition because it works well with numeric features and can handle noisy data. In speech, it uses MFCCs, pitch, and energy. In text, it uses embeddings or sentiment values. In facial recognition, it uses spatial features extracted from images.

Key Feature in Speech: MFCC

- **Mel Frequency Cepstral Coefficients (MFCCs):**

MFCCs are special features that capture how humans actually hear sound. They represent pitch, tone, and timbre. By breaking down speech into small frames and analyzing frequencies, MFCCs provide meaningful patterns that help models like RNNs, LSTMs, and CNNs detect if a person sounds happy, angry, or sad.

IMPLEMENTATION

1. Integration into Flask Application

Text Route

@app route ('/text', methods=['POST'])
Load the vectorizer and text model.
Retrieve user input text from the form.
Preprocess and vectorize the input text.
Predict emotion using the loaded model.
Return the predicted emotion label to the template.

In a Flask application, the /text route is defined to handle user-submitted text data for emotion recognition. This route listens for **POST** requests, indicating that it processes data sent from a form. When the route is accessed, the previously trained **text vectorizer** and **text model** are loaded to ensure compatibility with the preprocessing steps and the trained model. The user input text is retrieved from the submitted form data using Flask's request handling methods, such as request.form.get(). Next, the input text is preprocessed to match the format expected by the model. This typically involves cleaning the text (e.g., removing punctuation or converting to lowercase) and vectorizing it using the loaded vectorizer, which transforms the raw text into a numerical representation compatible with the model. The vectorized input is then passed to the trained model, which predicts the associated emotion. The predicted emotion label is extracted from the model's output and returned to the user through a rendered template, providing an interactive and real-time text-based emotion detection experience.

Image Route

@app.route ('/image', methods=['POST'])
Load the image model.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 9, September 2025

DOI: 10.17148/IJARCCE.2025.14928

Save the uploaded image file to a local path.
Preprocess the image (resize, normalize).
Predict emotion using the loaded model.
Return the predicted emotion label and image path to the template.

2. Integration into Flask Application

Text Route

@app route ('/text', methods=['POST'])
Load the vectorizer and text model.
Retrieve user input text from the form.
Preprocess and vectorize the input text.
Predict emotion using the loaded model.
Return the predicted emotion label to the template.

In a Flask application, the /text route is defined to handle user-submitted text data for emotion recognition. This route listens for **POST** requests, indicating that it processes data sent from a form. When the route is accessed, the previously trained **text vectorizer** and **text model** are loaded to ensure compatibility with the preprocessing steps and the trained model. The user input text is retrieved from the submitted form data using Flask's request handling methods, such as request.form.get(). Next, the input text is preprocessed to match the format expected by the model. This typically involves cleaning the text (e.g., removing punctuation or converting to lowercase) and vectorizing it using the loaded vectorizer, which transforms the raw text into a numerical representation compatible with the model. The vectorized input is then passed to the trained model, which predicts the associated emotion. The predicted emotion label is extracted from the model's output and returned to the user through a rendered template, providing an interactive and real-time text-based emotion detection experience.

Image Route

@app.route ('/image', methods=['POST'])
Load the image model.
Save the uploaded image file to a local path.
Preprocess the image (resize, normalize).
Predict emotion using the loaded model.
Return the predicted emotion label and image path to the template.

In a Flask application, the /image route is set up to handle image-based emotion recognition. This route accepts **POST** requests, allowing users to upload image files through a form. When a request is received, the trained **image model** is loaded to ensure compatibility for inference. The uploaded image file is retrieved from the request and saved to a local path using Flask's file handling utilities, such as request.file() and .save(). Next, the saved image is preprocessed to prepare it for the model. This involves resizing the image to match the input dimensions required by the model and normalizing the pixel values, typically by scaling them to a range between 0 and 1. The preprocessed image is then passed through the loaded model, which predicts the emotion based on the input data. The resulting emotion label, along with the path to the uploaded image, is returned to a rendered template. This setup provides users with a seamless interface for uploading images and receiving real-time emotion predictions.

Audio Route

@app.route ('/audio', methods=['POST'])
Load the audio model.
Save the uploaded audio file to a local path.
Preprocess the audio file (extract features).
Predict emotion using the loaded model.
Return the predicted emotion label and confidence to the template.

In a Flask application, the /audio route handles requests for emotion recognition from audio files. This route accepts **POST** requests, allowing users to upload audio files through a form. Upon receiving a request, the trained **audio model** is loaded to ensure it is ready for inference. The uploaded audio file is retrieved using Flask's request.files and saved to a local directory using the save () method, providing a path for further processing.

Next, the audio file undergoes preprocessing to prepare it for prediction. Using the **librosa** library, features such as **MFCCs**, **chroma features**, **mel spectrogram**, **spectral contrast**, and **tonnetz** are extracted from the audio. These features are then stacked to form a comprehensive feature vector matching the input size required by the model. This feature vector is passed through the loaded model to predict the emotion. The model outputs probabilities for each emotion class, and the class with the highest probability is identified as the predicted emotion label. Additionally, the confidence score of the prediction is included. Both the emotion label and confidence score are returned to a rendered template,

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 9, September 2025

DOI: 10.17148/IJARCCE.2025.14928

providing users with an interactive way to analyze emotions in audio inputs.

3. Integration into Flask Application

Text Route

@app route ('/text', methods=['POST'])
Load the vectorizer and text model.
Retrieve user input text from the form.
Preprocess and vectorize the input text.
Predict emotion using the loaded model.
Return the predicted emotion label to the template.

In a Flask application, the /text route is defined to handle user-submitted text data for emotion recognition. This route listens for **POST** requests, indicating that it processes data sent from a form. When the route is accessed, the previously trained **text vectorizer** and **text model** are loaded to ensure compatibility with the preprocessing steps and the trained model. The user input text is retrieved from the submitted form data using Flask's request handling methods, such as request.form.get(). Next, the input text is preprocessed to match the format expected by the model. This typically involves cleaning the text (e.g., removing punctuation or converting to lowercase) and vectorizing it using the loaded vectorizer, which transforms the raw text into a numerical representation compatible with the model. The vectorized input is then passed to the trained model, which predicts the associated emotion. The predicted emotion label is extracted from the model's output and returned to the user through a rendered template, providing an interactive and real-time text-based emotion detection experience.

Image Route

@app.route ('/image', methods=['POST'])
Load the image model.
Save the uploaded image file to a local path.
Preprocess the image (resize, normalize).
Predict emotion using the loaded model.
Return the predicted emotion label and image path to the template.

In a Flask application, the /image route is set up to handle image-based emotion recognition. This route accepts **POST** requests, allowing users to upload image files through a form. When a request is received, the trained **image model** is loaded to ensure compatibility for inference. The uploaded image file is retrieved from the request and saved to a local path using Flask's file handling utilities, such as request.file() and .save(). Next, the saved image is preprocessed to prepare it for the model. This involves resizing the image to match the input dimensions required by the model and normalizing the pixel values, typically by scaling them to a range between 0 and 1. The preprocessed image is then passed through the loaded model, which predicts the emotion based on the input data. The resulting emotion label, along with the path to the uploaded image, is returned to a rendered template. This setup provides users with a seamless interface for uploading images and receiving real-time emotion predictions.

Audio Route

@app.route ('/audio', methods=['POST'])
Load the audio model.
Save the uploaded audio file to a local path.
Preprocess the audio file (extract features).
Predict emotion using the loaded model.
Return the predicted emotion label and confidence to the template.

In a Flask application, the /audio route handles requests for emotion recognition from audio files. This route accepts **POST** requests, allowing users to upload audio files through a form. Upon receiving a request, the trained **audio model** is loaded to ensure it is ready for inference. The uploaded audio file is retrieved using Flask's request.files and saved to a local directory using the save () method, providing a path for further processing.

Next, the audio file undergoes preprocessing to prepare it for prediction. Using the **librosa** library, features such as **MFCCs**, **chroma features**, **mel spectrogram**, **spectral contrast**, and **tonnetz** are extracted from the audio. These features are then stacked to form a comprehensive feature vector matching the input size required by the model. This feature vector is passed through the loaded model to predict the emotion. The model outputs probabilities for each emotion class, and the class with the highest probability is identified as the predicted emotion label. Additionally, the confidence score of the prediction is included. Both the emotion label and confidence score are returned to a rendered template, providing users with an interactive way to analyze emotions in audio inputs.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 9, September 2025

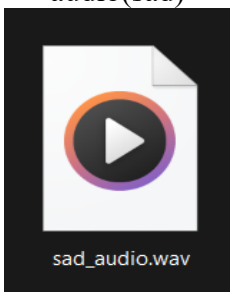
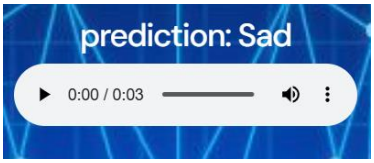
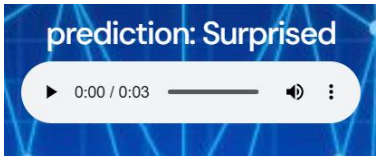
DOI: 10.17148/IJARCCE.2025.14928

SYSTEM TESTING, RESULTS AND DISCUSSION

Table 7.1.1: Unit Test Cases for Image Emotion and Audio recognition model

Test Case Number	Input	Stage	Expected Behaviour	Observed Behaviour	Status P=Pass F=Fail
1	(Happy) 	Preview page	Should display result as happy		P
2	(Angry) 	Preview Page	Should display result as angry		F

Table 7.1.2: Unit Test Cases for Audio Emotion recognition model

Test Case Number	Input	Stage	Expected Behavior	Observed Behavior	Status P=Pass F=Fail
1	audio(sad) 	Preview page	Should display result as sad		P
2	(Surprise)	Preview Page	Should display result as Surprise		P

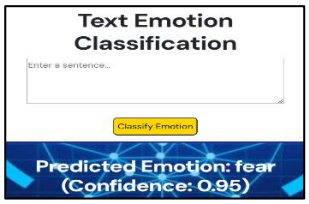
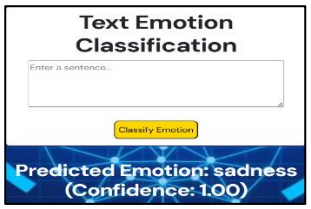
IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 9, September 2025

DOI: 10.17148/IJARCCE.2025.14928

Table 7.1.3: Unit Test Cases for Text Emotion recognition model

Test Case Number	Input	Stage	Expected Behavior	Observed Behavior	Status P=Pass F=Fail
1	(Fear) i have immense sympathy with the general point but as a possible proto writer trying to find time to write in the corners of life and with no sign of an agent let alone a publishing contract this	Preview page	Should display result as Fear		P
2	(Sadness) i didnt really feel that embarrassed	Preview Page	Should display result as Sadness		P

RESULT ANALYSIS

The main aim of the project was to classify the emotions using machine learning algorithms. Table 7.2 shows the analysis that was performed on the four algorithms with different training and testing sizes. It was found that LSTM was the most accurate in all the cases for text emotion recognition.

The result analysis is based on the accuracy of the four different algorithms used for emotion recognition models.

accuracy = (number of correct predictions) / (total number of predictions)

For example, if the model predicts the sentiment of 100 tweets as positive or negative, and 85 of them match the true sentiment, then the accuracy is 85/100 = 0.85 or 85%.

Table 7.2: Analysis of the four algorithms

Training Size	Testing Size	Accuracy (%)			
		CNN	RNN	LSTM	MFCC
70%	30%	87.89	85.60	89.92	84.78
60%	40%	86.62	84.78	88.32	83.54

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 9, September 2025

DOI: 10.17148/IJARCCE.2025.14928

Figure 7.2.2 shows the bar graph for the accuracy of the four algorithms where the train set size was 70% and the test set size was 30%.

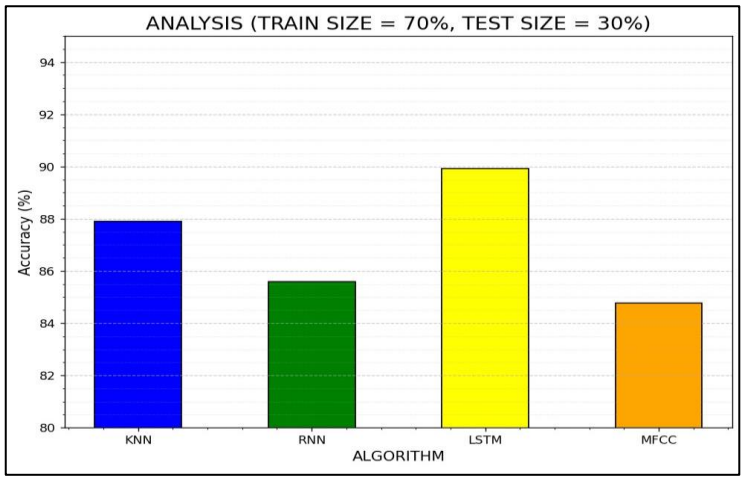


Figure 7.2.2: Graph analysis of the second set

Figure 7.2.3 shows the bar graph for the accuracy of the four algorithms where the train set size was 60% and the test set size was 40%.

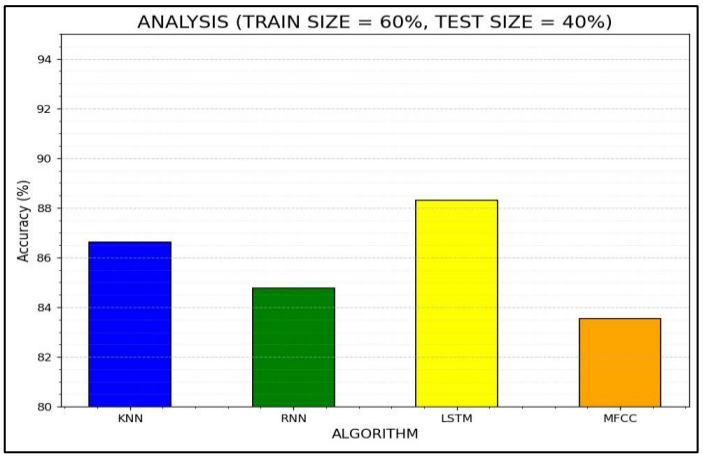


Figure 7.2.3: Graph analysis of the third set

CONCLUSION AND SCOPE FOR FUTURE WORK

Conclusion

The multimodal emotion classification system developed in this project effectively recognizes emotions across text, image, and audio modalities, offering a robust and comprehensive solution for emotion detection. Leveraging fine-tuned BERT for text analysis, CNN for image processing, and a CNN-based architecture for audio, the system achieves remarkable accuracy, with text emotion recognition reaching 91% and CNN models successfully classifying emotions from facial expressions and speech signals. Advanced feature-level fusion techniques combine speech features like pitch and MFCCs with facial embeddings, enabling synchronized and context-aware emotion detection. Transfer learning enhances the system's ability to generalize across diverse datasets, while real-time processing ensures scalability for live applications in areas such as affective computing, human-computer interaction, and mental health monitoring. Furthermore, the system incorporates advanced pre-processing pipelines to handle noisy data and uses cross-modal attention mechanisms to refine classification accuracy. Its versatility extends to applications in education, customer service, and entertainment, where understanding human emotions can provide personalized and adaptive experiences. Future work involves exploring Transformer-based multimodal fusion, integrating physiological signals such as heart rate and galvanic skin response for richer emotion analysis, and developing user-friendly interfaces for broader accessibility.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 9, September 2025
DOI: 10.17148/IJARCCE.2025.14928

Scope for Future Work

Future research in multimodal emotion classification could focus on improving fusion techniques (e.g., late, early fusion, and attention mechanisms) to enhance system performance. Incremental learning can help the model adapt to new emotional expressions over time, while cross-modal transfer learning may improve efficiency in training with limited labeled data. Real-time emotion recognition for applications like virtual assistants and emotion-sensitive interfaces is another promising area. Expanding to multilingual emotion recognition and applications in healthcare and education can address diverse contexts. Ethical considerations regarding privacy, fairness, and transparency will be crucial as these technologies evolve.

REFERENCES

[1]. Szegedy, Christian & Toshev, Alexander & Erhan, Dumitru. (2013). Deep Neural Networks for Object Detection. 1-9.

[2]. Benk, Sal & Elmir, Youssef & Dennai, Abdeslem. (2019). A Study on Automatic Speech Recognition. 10. 77-85. 10.6025/jitr/2019/10/3/77-85.

[3]. P. S. Sreeja and G. S. Mahalakshmi, ‘‘Emotion recognition from poems by maximum posterior probability,’’ Int. J. Comput. Sci. Inf. Secur., vol. 14, pp. 36–43, 2016.

[4]. J. Kaur and J. R. Saini, ‘‘Punjabi poetry classification: The test of 10 machine learning algorithms,’’ in Proc. 9th Int. Conf. Mach. Learn. Comput. (ICMLC), 2017, pp. 1–5.

[5]. G. Mohanty and P. Mishra, ‘‘Sad or glad? Corpus creation for Odia poetry with sentiment polarity information,’’ in Proc. 19th Int. Conf. Comput. Linguistics Intell. Text Process. (CICLing), Hanoi, Vietnam, 2018.

[6]. Y. Hou and A. Frank, ‘‘Analysing sentiment in classical Chinese poetry,’’ in Proc. 9th SIGHUM Workshop Lang. Technol. Cultural Heritage, Social Sci., Hum. (LaTeCH),15–24.

[7]. Goodfellow, Ian, Bengio, Yoshua, and Courville, Aaron. (2016). *Deep Learning*. MIT.

[8]. Hinton, Geoffrey, Osindero, Simon, and Teh, Yee-Whye. (2006). "A fast-learning algorithm for deep belief nets." *Neural Computation*, vol. 18, no. 7, pp. 1527–1554.

[9]. LeCun, Yann, Bengio, Yoshua, and Hinton, Geoffrey. (2015). "Deep learning." *Nature*, vol. 521, no. 7553, pp. 436–444.

