Smart Chat: A Mobile Chat Application Based on Machine Learning

Gunabalan Yathuraj
School of Computing,
SLIIT City Uni
yathu1015@gmail.com

Anuruddha Abeysinghe
School of Computing
SLIIT City Uni
anuruddha.a@sliit.lk

***Abstract* - In an increasingly digital world, communication is primarily conducted through messaging apps, but these platforms cannot often convey emotional nuance. This limitation can lead to misunderstandings, emotional disconnects, and deteriorating relationships. SmartChat addresses this gap by integrating machine learning-based emotion recognition into a mobile chat app, allowing users to send and receive voice messages enriched with emotional context. Built using React Native and compatible with both Android and iOS, SmartChat analyzes voice cues such as tone, pitch, and cadence to detect and display emotions to the user. This innovation improves the clarity and empathy of conversations, making digital communications more human-centered. Beyond general messaging, SmartChat has the potential to be used in critical contexts such as education, mental health support, and emotional literacy. By making emotionally aware communication accessible across languages and cultures, SmartChat contributes to fostering healthy interpersonal relationships and supports the broader goal of social sustainability through technology.**

***Keywords* - Machine Learning, Voice Analysis, Natural Language Processing, Emotion Detection**

I. INTRODUCTION

In today's digitally connected world, mobile communication platforms have become a necessity for personal, professional, and educational communication. However, these sites often lack emotional depth, especially in voice messaging, where tone and emotion are important. Without emotional context, users may misconstrue intent, leading to misunderstandings and weak relationships. This growing disconnect highlights the limitations of current chat applications in supporting emotionally rich communication.

While tools such as emojis and reaction buttons attempt to convey emotion, they fail to capture real-time emotional expression, especially for voice interactions. This challenge is even more important for individuals with communication barriers, such as those on the autism spectrum or speakers from diverse linguistic backgrounds. To create more inclusive and meaningful digital experiences, there is a growing demand for communication solutions that integrate emotional intelligence into their core design.

SmartChat addresses this need by introducing a cross-platform mobile chat application that uses machine learning to detect and display emotions in voice messages. Built with React Native and powered by a CNN trained on MFCCs, SmartChat recognizes emotional cues and displays them before acting on them. This approach enhances emotional clarity, empathy, and user connection. With potential applications in education, healthcare, military communication, and mental health, SmartChat redefines digital communication by embedding emotional awareness into everyday conversations.

II. LITERATURE REVIEW

The need to improve emotional intelligence in digital communication systems has led to the emergence of speech emotion recognition (SER) as a key research area in artificial intelligence and human-computer interaction. Early research by Scherer et al. using support vector machines (SVMs) to classify primary emotions through phonological and linguistic features had moderate success but faced limitations due to speaker variability [4]. Subsequent efforts introduced techniques such as Mel-Frequency Cepstral Coefficients (MFCCs) and Gaussian Mixture Models (GMMs) to improve emotion detection accuracy [1][3]. For example, Lee et al incorporated speaker normalization into their model and achieved an accuracy of 87.4% [2]. Recently, deep learning approaches have shown remarkable improvements in classification performance. Zhang et al. used spectrograms with convolutional neural networks (CNNs) to achieve an accurate rate of 75.1% [5], while Wu et al. combined CNNs and long-short-term memory (LSTM) networks to achieve an accuracy of over 84% [10]. These results underline the growing performance of deep learning in emotion recognition tasks [7][8][9].

Beyond acoustic modeling, researchers have also examined physiological and visual signals, such as heart rate variability (HRV), galvanic skin response (GSR), and

https://doi.org/10.54389/WMCR8688

EEG, for emotion detection. While promising, these methods are often invasive or require specialized hardware [6]. Meanwhile, multimodal approaches combining face recognition, speech tone, and linguistic cues have gained traction to improve SER accuracy [4][6]. Comparative studies have further explored SER in different linguistic and cultural contexts, with systems developed in regional languages such as Marathi and Romanian demonstrating cross-linguistic compatibility [1].

Domain-specific applications of SER continue to expand, including its use in emotionally aware customer service systems, autism diagnosis, and interactive entertainment platforms [3][5]. Despite technological advances, there remain major challenges in generalizing SER models across speakers, languages, and sensory nuances. Based on these fundamentals, SmartChat is built by applying CNN and MFCC-based emotion recognition to real-time mobile chat applications, contributing to the technological advancement and practical application of SER in everyday communication.

SmartChat addresses this gap by introducing a real-time, cross-platform mobile chat app that embeds SER into everyday voice communication. Its universal design – compatible across languages and suitable for users with communication challenges – distinguishes it as a socially inclusive and practical solution. Therefore, the unique contribution of SmartChat lies in combining real-time SER with mobile messaging, providing an emotionally intelligent communication tool for everyday life.

III. METHODOLOGY

SmartChat's development followed an agile methodology to ensure iterative improvement, continuous testing, and adaptability throughout the software lifecycle. This approach divided the project into multiple development sprints, allowing for frequent revisions based on user feedback and technical assessments. Each sprint included phases of planning, implementation, testing, and retrospective analysis, fostering a flexible and user-centric development process. The core system was developed using React Native for the mobile interface, while the backend logic, including machine learning processing, was developed in Python and deployed using Flask APIs. Firebase was used to manage authentication, cloud messaging, and real-time database interactions. For emotion detection, the team utilized a convolutional neural network (CNN) framework combined with Mel-Frequency Cepstral Coefficients (MFCCs) to extract meaningful acoustic features from voice messages. Model training and testing were conducted in Google Colab, using cloud-based GPU resources for efficient computation. The agile methodology ensured that new features were regularly integrated, bugs were resolved quickly, and the final product was aligned with both technical requirements and user expectations. This structured but flexible approach was essential to managing the complexity of combining real-time mobile communications with machine learning-based emotion recognition.

A. System Architecture & Features

SmartChat is built using a modular framework that combines React Native for a cross-platform mobile interface and Python with Flask for backend processing. Firebase handles authentication, real-time messaging, and cloud storage. The core feature is a speech emotion recognition engine that uses CNN and MFCCs to analyze voice messages and classify emotions such as happiness, anger, or sadness. These emotions are visualized in the chat interface before playback. Key features include real-time emotion detection, multilingual support, group chats, secure login, and a user-friendly design, making SmartChat an intuitive and inclusive platform for emotionally aware digital communication. As shown in Fig 1, the voice recognition module includes stages for audio capture, feature extraction, and classification.

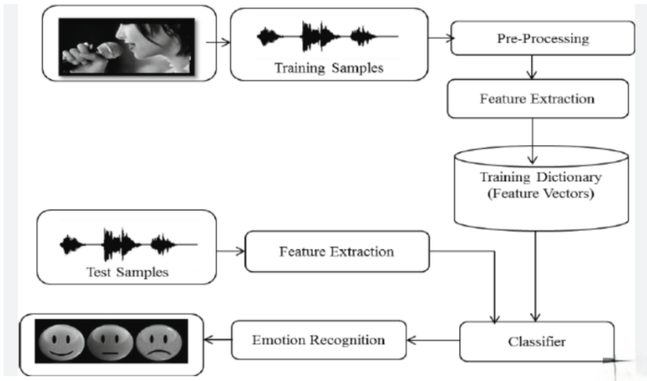

Fig 5 Model for Emotion Detection

B. Analysis

SmartChat comprises four main components that work together to deliver emotion-aware communication. The front-end is built using React Native, allowing for cross-platform compatibility for both Android and iOS devices. The back end is built with Python and Flask, handling API requests and connecting to a machine learning model. Firebase serves as cloud infrastructure, securely managing real-time messaging, user authentication, and data storage. The emotion recognition module is the heart of the system, classifying emotions in real-time using a convolutional neural network (CNN) trained on MFCC-processed voice data. Together, these components enable SmartChat to deliver responsive, accurate, and emotionally intelligent messaging experiences. The tools and technologies used in the development of *FocusBoost* are summarized in Table 1.

Table 2 Tools and Technologies

Frontend	React Native
Backend	Python, Flask
Database & Authentication	Firebase
Machine Learning Model	CNN
Feature Extraction	MFCC
Development Tools	Visual Studio Code, GitLab
Platforms	Android, iOS

C. Key Components

- React to Native Frontend:
  Provides a unified, cross-platform user interface for Android and iOS devices. It enables users to register, log in, send text and voice messages, and visualize detected emotions in real time.
- Flask API (Python Backend):
  It handles RESTful API requests for message processing and emotion detection.
- Firebase Services:
  Facilitates real-time messaging, secure user authentication, and cloud storage of chat data.
- Machine Learning Model:
  Implements a Convolutional Neural Network (CNN) trained on MFCC-processed voice data.

IV. DEVELOPMENT & DISCUSSION

The development of SmartChat followed an agile approach, which enabled continuous iteration and feature refinement throughout the project lifecycle. The application was built using a combination of React Native for the user interface and Python with Flask for backend logic. Firebase was integrated to handle real-time messaging and user authentication, ensuring both scalability and security. A machine learning model built using a CNN framework was trained on a custom-labeled voice dataset with MFCC features to accurately classify emotional states. Throughout development, several test sprints were conducted to validate the system's functionality, including login authentication, real-time voice message transmission, and accurate emotion detection. During testing, the emotion classification model achieved 71% accuracy, indicating promising performance for a real-time mobile application. Despite this success, there were challenges in ensuring the model's generalization across a variety of accents, background noise, and various emotional expressions. These limitations suggest potential for future improvements through larger datasets, improved pre-processing, and multilingual emotion training. Overall, the development of SmartChat illustrates the feasibility and social value of integrating emotion-aware AI into mobile communication platforms.

SmartChat's emotion detection system uses a convolutional neural network (CNN) trained on Mel-Frequency Cepstral Coefficients (MFCCs), which captures key audio features such as pitch and tone. The framework includes two convolutional layers (32 and 64 filters, 3×3 kernel) with ReLU implementation, each reducing spatial dimensions by 2×2 max-pooling layers. The output is flattened and passed through 128 ReLU-enabled neurons and a dense layer with a dropout layer of 0.5 to prevent overfitting. The model is trained using the Adam optimizer and categorical cross-entropy loss, stopping early to improve generalization.

V. TESTING AND EVALUATION

The testing and evaluation phase of SmartChat was critical to ensuring the system's stability, usability, and accuracy in delivering real-time emotion-aware messages. A structured test plan was developed that included functional and non-functional test cases. The functional tests evaluated key functionalities such as user registration, login authentication, sending and receiving text and voice messages, and integration of the emotion recognition feature. Each test case included specific input scenarios – such as entering valid and invalid credentials, submitting empty fields, and attempting to log in with incomplete data – to ensure that the application responded appropriately with validations and error messages. These tests demonstrated that the front-end and back-end workflows were properly synchronized and responsive to multiple user actions.

The focus of the evaluation was the performance of the emotion recognition model. Voice samples with known emotional tones – such as anger, sadness, happiness, and neutrality – were processed through the system to verify that the CNN-MFCC model could correctly classify them. The model achieved an accurate rate of approximately 71%, which is remarkable for a mobile-based real-time application, considering constraints such as device processing power and varying acoustic environments. The evaluation also revealed that while the model performed well with clearly expressed emotions and consistent speech patterns, it struggled somewhat with accents, background noise, and subtle emotional variations.

Non-functional testing focused on system performance, usability, and reliability. Users evaluated the app for responsiveness, interface design, and ease of use. Feedback was largely positive, highlighting the app's intuitive navigation, emotional visualization features, and clean design. Users appreciated the visual cues associated with voice messages, which added emotional context and enriched the interaction experience. However, users also noted the need for more detailed feedback for additional emotion types and ambiguous tones. Load testing under limited conditions showed stable performance with multiple users, but scalability testing on a larger network was identified as future work. Figure 2 presents the workflow of emotion detection, which plays a key role in analyzing user responses and adapting the content accordingly.

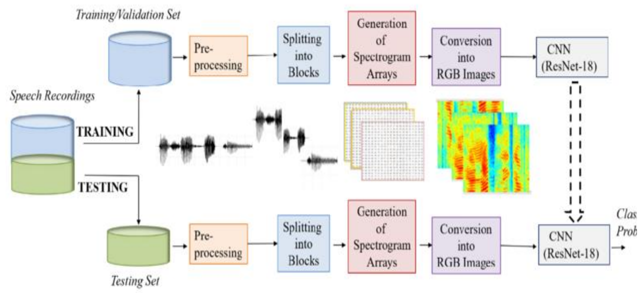

Fig 6 CNN model training and testing framework

Overall, the testing and evaluation phase confirmed that SmartChat meets its operational requirements and performs reliably in recognizing and presenting emotions in voice communications. These findings confirm the reliability of the project while also identifying valuable insights for future improvements, such as increasing model robustness, expanding emotion classifications, and improving support for diverse linguistic and cultural contexts.

VI. CONCLUSION & FUTURE WORK

A. *Conclusion*

SmartChat demonstrates the potential of integrating machine learning with mobile communication to improve emotional awareness and digital empathy. This application successfully combines a user-friendly mobile interface with a real-time speech emotion recognition engine built on CNN and MFCC technologies. During testing, the emotion classification model achieved 71% accuracy, confirming its effectiveness in detecting key emotional states such as happiness, sadness, anger, and neutrality in voice messages. The system reliably recognized emotions and displayed them clearly to users before audio playback, enabling more emotionally intelligent interactions.

Key findings of the project highlight the use, accessibility, and relevance of SmartChat in a variety of communication situations. The application proved particularly valuable for individuals with communication challenges and in contexts such as distance learning, telehealth, and long-distance relationships. Users responded positively to its intuitive design and appreciated its unique ability to provide emotional context in digital conversations – an element often missing from traditional chat platforms.

Overall, SmartChat provides a new solution to the growing need for emotionally aware technology. By enabling real-time emotion detection in everyday messaging, it improves user engagement, reduces miscommunication, and enhances emotional well-being.

B. *Future Work*

While SmartChat has proven its potential as an innovative emotion-aware communication platform, there are still many avenues for future development. Additionally, integrating facial emotion recognition into video chats and improving sentiment analysis into text messages will enable a more complete understanding of user emotions. Future updates will also include user analytics to track emotional trends over time, which could support personal well-being or therapeutic use cases. Ultimately, these improvements aim to make SmartChat a more powerful, adaptive, and supportive tool for fostering emotionally rich digital communication.

ACKNOWLEGEMENT

I sincerely thank my supervisor for his valuable guidance and support throughout this research. Special thanks to those who helped me access key resources and datasets. I deeply appreciate the unwavering support of all.

REFERENCES

[1] Al-Talabani, A. A. (2018). Speech Emotion Recognition using Machine Learning Techniques: A Review. Journal of Telecommunication, Electronicand Computer Engineering, 10(1-11), 101- 105.

[2] Kaushik, S., & Bhatia, S. (2019). Speech Emotion Recognition: A Comprehensive Study of Machine Learning Techniques. In Proceedings of 3rd International Conference on Communication System, Computing and IT Applications (pp. 1-6).

[3] Kim, K., Lee, K., & Kim, J. (2019). Emotion Recognition from Speech Using Machine Learning Techniques. Journal of the Korean Society for Speech Sciences, 11(1), 1-7.

[4] El Ayadi, M., Kamel, M. S., & Karray, F. (2011). Survey on speech emotion recognition: Features, classification schemes, and databases. Pattern Recognition, 44(3), 572-587.

[5] Zhang, Y., Liu, X., & Chen, S. (2019). Emotion Recognition from Speech Using Ensemble Machine Learning Techniques. In Proceedings of the 6th International Conference on Education, Language, Art and Inter-cultural Communication (pp. 305-310).

[6] Eyben, F., Schuller, B., Wöllmer, M., & Rigoll, G. (2010). Emotion recognition from speech: a review. In Proceedings of the 18th ACM International Conference on Multimedia (pp. 4-15).

[7] Mollahosseini, A., & Chan, D. (2017). Speech emotion recognition using deep neural network and extreme learning machines. In Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 223-227).

[8] Yang, S., Hu, Z., & Chen, J. (2018). An emotional speech recognition algorithm based on deep learning. In Proceedings of the 2018 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC) (pp. 51-56).

[9] Aghanavesi, S., Heydarian, P., & Ahmadi, H. (2019). A Deep Learning Approach for Speech Emotion Recognition. In Proceedings of the 2019 6th International Conference on Signal Processing and Integrated Networks (SPIN) (pp. 672-676).

[10] Wu, Y., Chu, X., & Huang, X. (2018). Emotion recognition from speech using convolutional neural network and long short-term memory. In Proceedings of the 2018 IEEE International Conference on Big Data and Smart Computing (BigComp) (pp. 95-100).