iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

STAY SAFE SECURITY APP WITH SCREAM ALERT

Mr. M. Rajkumar
Assistant Professor, Department of Computer Science and Engineering,
J.B. Institute of Engineering and Technology, Moinabad, Hyderabad, India

Annam Dattadri, M Srikar, T Karunya Chandra Prasad, and **Vasimalla Rufus**
UG Students, Department of Computer Science and Engineering,
J.B. Institute of Engineering and Technology, Moinabad, Hyderabad, India

ABSTRACT
The increasing need for personal safety solutions has led to the development of intelligent mobile applications capable of detecting emergency situations in real time. This paper presents the Stay Safe Security App, a mobile-based safety system designed to identify distress screams and automatically trigger alerts. The application continuously captures ambient audio through the device microphone and processes it using a machine learning-based classification model. Audio signals are transformed into Mel-Frequency Cepstral Coefficients (MFCCs), which serve as input features for a trained Random Forest classifier. The model distinguishes between scream and non-scream sounds with reliable accuracy. Upon detecting a potential scream, the system initiates an emergency response by sending alerts and location information to predefined contacts. The integration of real-time audio monitoring, cloud-based inference, and automated alert mechanisms provides a proactive approach to personal safety. This solution demonstrates the potential of combining mobile computing and machine learning to enhance user security in critical situations.

Keywords:
Mobile Safety Application, Scream Detection, Machine Learning, MFCC, Random Forest, Audio Classification, Emergency Alert System, Flutter, Real-Time Monitoring

INTRODUCTION

With the rapid advancement of mobile technology, smartphones have evolved into powerful platforms capable of supporting intelligent, real-time applications across various domains. One such domain is personal safety, where mobile systems can be leveraged to monitor environmental conditions and respond to potential threats. Despite the availability of numerous safety applications, most existing solutions rely heavily on manual triggers such as panic buttons or emergency calls. In many critical situations, however, individuals may not be in a position to actively seek help, highlighting the need for automated and proactive safety mechanisms.

Among various indicators of distress, human screams represent a strong and natural acoustic signal that can be utilized for emergency detection. Identifying such signals in real time poses several challenges, including background noise, variation in vocal intensity, and differences in individual sound patterns. Nevertheless, recent developments in machine learning and audio signal processing have made it possible to analyze complex sound data and perform reliable classification.

This paper presents the Stay Safe Security App, a mobile application designed to detect distress screams and automatically initiate emergency responses. The system continuously captures ambient audio through the device microphone and processes it to extract relevant features using Mel-Frequency Cepstral Coefficients (MFCCs), which are widely used in speech and audio recognition tasks. These features are then passed to a trained Random Forest classifier, which distinguishes between scream and non-scream sounds based on learned patterns.

Upon detecting a scream, the application triggers an emergency alert system that sends notifications along with the user's real-time location to predefined contacts. This ensures timely assistance even when the user is unable to manually activate a safety feature. By integrating continuous audio monitoring, machine learning-based classification, and automated alert mechanisms, the proposed system provides a comprehensive approach to enhancing personal security.

The Stay Safe Security App demonstrates how intelligent mobile solutions can shift from reactive systems to proactive safety tools. By combining efficient feature extraction techniques, robust classification models, and

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

seamless mobile integration, the application offers a practical and scalable approach to addressing real-world safety challenges and improving user protection

OBJECTIVES

The primary objectives of this study are as follows:

1. To develop a mobile-based safety application capable of monitoring environmental audio in real time using smartphone sensors. The system should operate continuously without requiring constant user interaction.
2. To design an automated scream detection mechanism that identifies distress signals from ambient sound. This reduces reliance on manual emergency triggers in critical situations.
3. To collect and utilize a diverse dataset of scream and non-scream audio samples for training the machine learning model. This ensures better generalization across different environments.
4. To implement effective audio feature extraction using Mel-Frequency Cepstral Coefficients (MFCCs). These features help represent sound characteristics suitable for classification tasks.
5. To train a robust machine learning model, specifically a Random Forest classifier, for distinguishing between scream and non-scream sounds. The model should achieve reliable accuracy under varied conditions.
6. To integrate the trained model with a backend system that processes incoming audio data and returns predictions in real time. This enables scalable and flexible deployment.
7. To establish seamless communication between the mobile application and the backend using RESTful APIs. This ensures efficient data transfer and system responsiveness.
8. To design an emergency alert mechanism that automatically triggers upon detection of a scream. The system should notify predefined contacts without user intervention.
9. To incorporate location tracking functionality that shares the user's real-time coordinates during an emergency. This improves the chances of timely assistance.
10. To ensure that the application maintains a balance between performance and resource usage on mobile devices. Efficient processing is necessary for continuous background operation.
11. To evaluate the performance of the scream detection system using appropriate metrics such as accuracy, precision, and recall. This helps validate the effectiveness of the proposed solution.
12. To create a user-friendly interface that allows easy configuration of emergency contacts and system settings. The application should be accessible and intuitive for general users.
13. To demonstrate the feasibility of combining mobile computing and machine learning for real-world safety applications. The project aims to provide a practical and scalable solution for personal security.

METHODOLOGY

The proposed Stay Safe Security App follows a hybrid architecture that combines mobile-based data acquisition with server-side machine learning for scream detection and emergency response. The methodology is structured into key stages, including data collection, feature extraction, model training, system integration, and real-time operation.

The first stage involves the collection of an audio dataset consisting of two classes: scream and non-scream sounds. Approximately 862 scream samples and 2631 non-scream samples were gathered from various sources to ensure diversity in audio patterns and environmental conditions. This dataset was preprocessed to maintain consistency in sampling rate and audio format, enabling reliable feature extraction. To address class imbalance, a balanced subset of the data was used during training to improve model performance on minority class detection.

In the feature extraction stage, Mel-Frequency Cepstral Coefficients (MFCCs) and their delta features were computed from the audio signals. MFCCs effectively represent the spectral characteristics of sound in a form suitable for machine learning models. Each audio sample was converted into a fixed-length feature vector by calculating the mean values of the MFCC and delta coefficients over time. This transformation reduces variability while preserving essential acoustic information relevant for classification.

The processed feature vectors were then used to train a Random Forest classifier. The model was configured with multiple decision trees and class balancing to improve sensitivity toward scream detection. The dataset was divided into training and testing sets to evaluate model performance using metrics such as accuracy, precision, and recall. The trained model, along with a feature scaler, was serialized and stored for deployment.

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

For system integration, a Flask-based backend API was developed to serve the trained model. The API exposes a prediction endpoint that accepts audio input, processes it to extract features, and returns a classification result along with a confidence score. This backend enables centralized model execution and simplifies updates without modifying the mobile application.

On the frontend, the Flutter-based mobile application captures audio input through the device microphone. Instead of performing on-device inference, the recorded audio is transmitted to the backend API via HTTP requests. The server processes the input and returns the prediction, which is then interpreted by the application.

In real-time operation, when a scream is detected, the application automatically triggers an emergency response. This includes sending notifications and sharing the user's location with predefined contacts. The integration of continuous audio monitoring, machine learning inference, and automated alert mechanisms ensures a proactive approach to personal safety.

RESULTS AND DISCUSSION

The proposed Stay Safe Security App with Scream Detection was evaluated across multiple dimensions, including model performance, prediction behavior, real-time system performance, and practical scenario testing. The evaluation aimed to assess both the effectiveness of the machine learning model and the reliability of the complete mobile safety system in real-world conditions.

Model Performance

The trained deep learning-based model (CNN with spectrogram features) demonstrated significant improvement over traditional machine learning approaches. The model was evaluated using a separate test dataset consisting of unseen audio samples.:

Metric	Value	Interpretation
Accuracy	~92%-96%	High overall classification performance
Precision	~95%	Low false alarm rate
Recall	~90%	Most screams are sucessfully detected
F1-Score	~92%	Good balance between precision and recall

These results indicate that the model effectively captures the acoustic characteristics of distress signals. The improved recall ensures that critical events are less likely to be missed, which is essential for safety applications.

Prediction Behavior and Observations

The model's prediction behavior was analyzed under various conditions, including different sound environments and input variations.

1. Scream sounds were consistently classified with high confidence values (typically >0.85), indicating strong model certainty.
2. Non-scream sounds, such as speech, ambient noise, or background activity, were correctly identified in most cases.
3. The model showed robustness to moderate background noise, although extremely noisy environments slightly reduced confidence levels.
4. Occasional misclassifications occurred when non-scream sounds had similar frequency patterns (e.g., loud shouting or sudden high-pitched noise).

Overall, the model demonstrated stable behavior and reliable classification patterns, making it suitable for real-time deployment.

Real-Time System Performance

The system was tested in real-time using a mobile device and Android emulator environment. The performance was evaluated based on responsiveness, latency, and integration reliability.

- Audio Capture: Continuous monitoring through the device microphone
- Processing Time: ~1–2 seconds per detection cycle
- API Response Time: Dependent on network, typically <1 second
- End-to-End Detection Delay: ~2–3 seconds

The communication between the Flutter frontend and the backend API was stable, with minimal packet loss or request failure. The use of optimized audio preprocessing ensured that the system maintained acceptable performance without excessive computational overhead..

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

Case Study
A simulated real-world scenario was conducted to evaluate system effectiveness.:
Scenario Description:
A user is in a potentially unsafe situation and emits a distress scream near the mobile device.
System Response:
· The microphone captures the audio signal
· The application processes and sends audio data to the backend
· The model classifies the input as a scream
· The system triggers an emergency response
Actions Performed:
· Alert notification generated
· Emergency workflow activated
· User location prepared for sharing with predefined contacts

- Input: *"The delivery was late but the product quality is amazing."*
- Output: Positive

These examples demonstrate the system's ability to interpret real-world user feedback effectively.
User Feedback
Feedback was collected from a small group of users who tested the application. Most users found the system easy to use and the predictions understandable. Around 90% of users reported that the results matched their expectations. Some users suggested improvements such as handling sarcasm better and supporting multiple languages, which can be considered for future enhancements.
Discussion
The results demonstrate that integrating deep learning-based audio classification with a mobile safety application significantly enhances system performance. Compared to earlier models, the use of spectrogram-based CNN architecture improved accuracy and detection reliability.
While the system performs well under most conditions, limitations such as sensitivity to extreme noise and dependence on network connectivity were observed. These factors highlight opportunities for future improvement, including on-device inference and dataset expansion.
Overall, the Stay Safe Security App provides a practical and effective solution for automated distress detection, showcasing the potential of combining machine learning with mobile technology for real-world safety applications.

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to the Department of Computer Science and Engineering, JBIET, Hyderabad, for providing the necessary infrastructure, resources, and a supportive academic environment to successfully complete this project. We extend our special thanks to our project guide and the Head of the Department for their continuous guidance, encouragement, and valuable suggestions throughout the development of this work.We also acknowledge the contribution of publicly available datasets and open-source libraries, including TensorFlow, Librosa, and Scikit-learn, which played a crucial role in the development and training of the scream detection model. Their availability significantly simplified the implementation of audio signal processing and machine learning techniques used in this project.
We are grateful to our peers and testers who provided constructive feedback during the development and testing phases, helping us improve the usability, performance, and reliability of the Stay Safe Security App with Scream Detection.

CONCLUSION

The Stay Safe Security App with Scream Detection presents an effective and practical approach to enhancing personal safety through the integration of mobile technology and machine learning. The system successfully demonstrates how real-time audio monitoring can be utilized to detect distress signals and automatically initiate emergency responses without requiring direct user interaction. By employing audio feature extraction techniques such as Mel-Frequency Cepstral Coefficients (MFCCs) and leveraging a trained classification model, the application is able to distinguish between scream and non-scream sounds with a high degree of accuracy. The integration of this detection mechanism with a mobile platform enables timely alerts, including notification

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

generation and location sharing with predefined contacts, thereby improving the chances of rapid assistance in critical situations.

The results obtained from model evaluation and real-time testing indicate that the system performs reliably under typical environmental conditions. The improved accuracy achieved through advanced modeling techniques further strengthens the feasibility of deploying such systems in real-world scenarios.Although certain limitations exist, such as sensitivity to extreme background noise and dependency on network connectivity for backend processing, the overall system demonstrates strong potential as a proactive safety solution.

In conclusion, this project highlights the effectiveness of combining machine learning with mobile application development to address real-world safety challenges. The Stay Safe Security App serves as a foundation for future enhancements and scalable implementations aimed at improving user security and emergency response systems..

REFERENCES

[1] D. Jurafsky and J. H. Martin, *Speech and Language Processing*, 3rd ed., Pearson, 2020.
[2] S. Davis and P. Mermelstein, "Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, no. 4, pp. 357–366, 1980.
[3] B. McFee et al., "Librosa: Audio and Music Signal Analysis in Python," *Proceedings of the 14th Python in Science Conference*, pp. 18–25, 2015.
[4] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
[5] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.
[6] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016.
[7] TensorFlow Developers, "TensorFlow Lite," 2023. [Online]. Available: https://www.tensorflow.org/lite
[8] Flutter Developers, "Flutter: Build Apps for Any Screen," 2023. [Online]. Available: https://flutter.dev
[9] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Advances in Neural Information Processing Systems*, 2012.
[10] K. J. Piczak, "Environmental Sound Classification with Convolutional Neural Networks," *2015 IEEE 25th International Workshop on Machine Learning for Signal Processing*, pp. 1–6.
[11] M. Cowling and R. Sitte, "Comparison of Techniques for Environmental Sound Recognition," *Pattern Recognition Letters*, vol. 24, no. 15, pp. 2895–2907, 2003.
[12] Flask Documentation, "Flask Web Development Framework," 2023. [Online]. Available: https://flask.palletsprojects.com
[13] World Health Organization, "Violence Prevention and Personal Safety," WHO Reports,