International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

AI Powered Scream Detection for Smart Emergency Alerts

Udhayan D, Vaseekaran A

Abstract

Emergency situations often require immediate attention, but detecting distress sounds in real-time remains a challenge. This paper presents an AI-powered scream detection system designed for smart emergency alerts. Using a Support Vector Machine (SVM) classifier trained on the ESC-50 dataset, the model distinguishes between scream-like sounds and non-emergency noises by extracting Mel-Frequency Cepstral Coefficients (MFCC) from audio signals.

The system is integrated into a Streamlit-based application that allows real-time audio recording and analysis. Upon detecting a scream, the system triggers an automated email alert, enhancing safety measures in smart environments. The proposed solution demonstrates high accuracy in distinguishing alert sounds, making it a viable tool for public security, surveillance, and healthcare applications.

Keywords: Sound Detection, Audio Classification, Support Vector Machine (SVM), MFCC Features, Emergency Sound Recognition, Audio Signal Processing, Human Alert System, Machine Learning, Sound-Based Alert, Streamlit Interface

1. Introduction

In recent years, the demand for intelligent and responsive safety systems has grown significantly, particularly in environments where human supervision is limited or delayed. Among various sensory modalities, sound has emerged as a powerful and often underutilized signal for identifying emergency situations. Sounds such as screams, alarms, or calls for help often indicate distress or danger and can serve as early warnings to prevent incidents or enable quicker responses.

This research focuses on the development of a **Human Alert Sound Detection System** capable of recognizing scream-like sounds in **real-time and triggering automated email alerts**. The system is designed to process both uploaded and live-recorded audio, extract relevant features using **Mel Frequency Cepstral Coefficients (MFCC)**, and classify the audio using a **Support Vector Machine (SVM) classifier**. If a scream is detected, the system not only displays the result but also sends an automated email notification, making it suitable for security monitoring, surveillance, and smart environment applications.

Built using Python and libraries such as Librosa, Scikit-learn, and Streamlit, the system ensures a modular, scalable, and user-friendly interface for real-time monitoring. Integration with SMTP email services and local audio recording tools enhances its utility and applicability in various domains, including smart homes, public safety infrastructure, and assistive technologies for the elderly or vulnerable populations.
The primary objective of this work is to develop a lightweight, reliable, and interpretable solution for detecting distress sounds, with minimal hardware requirements and high portability.

This paper details the system design, dataset selection, feature engineering, model development, and end- to-end implementation, providing a comprehensive overview of a deployable alert detection solution.

We focuses on detecting human alert sounds—specifically screams—from short audio clips recorded or uploaded by the user.

The key components within scope include:
- Audio preprocessing and MFCC feature extraction.
- SVM-based classification of audio inputs.
- A responsive UI with audio recording/upload capability.
- A real-time email alert mechanism.

The primary contributions of our work are:

- Real-time continuous audio streaming and monitoring.
- Multi-class audio classification beyond binary alert/non-alert.
- Integration with video or hardware surveillance systems.

2. Related Work

In recent years, audio-based detection systems have garnered increasing attention due to their applicability in security, surveillance, and assistive technologies. Several studies have explored the classification of environmental and emergency sounds using machine learning techniques.

Huang et al. (2010) investigated the use of Mel-Frequency Cepstral Coefficients (MFCCs) combined with Hidden Markov Models (HMMs) for identifying various environmental sounds. Their work laid the foundation for using MFCCs as a primary feature in sound classification due to their ability to capture perceptually relevant frequency components.

Cowling and Sitte (2003) compared multiple audio features and classifiers for environmental sound recognition. They concluded that MFCCs, when used with Support Vector Machines (SVMs), yielded high accuracy in distinguishing between different types of sounds, including alarms and screams.

Dennis et al. (2011) proposed a Gaussian Mixture Model (GMM)-based approach for sound event detection and emphasized the importance of temporal feature dynamics in classification. However, their system required high computational resources and was less effective in real-time applications.

Zhang et al. (2015) focused on scream and gunshot detection in surveillance systems using deep learning models. While these models achieved high precision, they often lacked efficiency in deployment on lightweight systems due to their complexity and need for large datasets.

Compared to deep learning approaches, SVMs provide a computationally efficient alternative for binary classification tasks such as scream detection, especially when training data is limited. Furthermore, the integration of MFCC features with SVMs offers a balance between accuracy and processing speed, making it suitable for real-time applications on limited-resource platforms.

Our work builds on these foundations by implementing a lightweight scream detection system using SVMs, MFCC feature extraction, and a user-friendly web interface using Streamlit. Additionally, we incorporate an automated email alert mechanism, expanding the practical utility of the system in real-world emergency scenarios.

3. Methodology

3.1 System Architecture

The proposed system is designed to detect human alert sounds (specifically screams) and respond with appropriate notifications. The architecture includes the following components:

- **Audio Input Interface**: Allows users to record or upload audio through a web interface.
- **Feature Extraction Module**: Extracts Mel Frequency Cepstral Coefficients (MFCC) from the input audio signal.
- **SVM Classifier Engine**: Uses a trained Support Vector Machine model to classify the audio as "scream" or "not a scream".
- **Alert Dispatcher**: Sends an email alert if a scream is detected.
- **Result Display Layer**: Presents the classification outcome to the user in real time.

3.2 Feature Extraction Strategy

To effectively represent audio data in a machine-learnable format, we implement the following:

3.2.1 **MFCC Transformation**: Converts the raw waveform into compact features that closely model human auditory perception.

3.2.2 **Preprocessing Steps**:

- Normalization of input amplitude.
- Noise filtering (optional for cleaner inputs).
- Frame segmentation and windowing for time-based analysis.

3.3 Classification Model

A supervised machine learning approach is adopted using Support Vector Machines (SVM):

Trained SVM Classifier:

- Kernel: Radial Basis Function (RBF)
- Hyperparameters optimized through cross-validation
- Binary classification: *Scream* vs. *Not a Scream*

3.4 Alerting and Integration Layer

The alert system is designed to be efficient and modular:

Email Alert System:

- Uses SMTP protocol for sending emails.
- Triggered only when a "scream" classification is made.

Streamlit-based UI:

- Enables audio upload/recording.
- Displays prediction results and system status to the user.

4. Implementation and Result

4.1 Implementation overview

To validate the proposed scream detection system, a working prototype was developed using Python 3.10, leveraging various audio processing and machine learning libraries. The front-end was built using Streamlit for ease of interaction and demonstration. The system was locally deployed and tested in both development and user-simulated environments.

The major components of the implementation include:

- **Audio Input Interface:** Implemented via Streamlit allowing users to either record audio in real-time or upload pre-recorded clips in .wav format.
- **Feature Extraction Module:** Utilizes librosa to extract MFCCs (Mel Frequency Cepstral Coefficients) from the audio input. Additional features like chroma and spectral contrast were evaluated during testing for model tuning.
- **Classification Model:** An SVM (Support Vector Machine) model with an RBF kernel was trained on pre-labeled datasets to detect whether the input audio contains a scream. The model was fine-tuned using scikit-learn with stratified cross-validation.
- **Email Alert System:** Integrated via the smtplib module. If the model predicts a scream, an alert email is automatically dispatched to a configured recipient.
- **Visualization and Feedback:** Classification results and logs are displayed on the Streamlit interface for user feedback. Real-time status updates are shown throughout the process.

4.2 Dataset and Testing

The system was trained and evaluated on three audio datasets:

- **Dataset A**: Urban Sound 8K – Contains a mix of environmental and human sounds (~8,732 audio clips).
- **Dataset B**: Custom Scream Collection – 300+ manually labeled scream and non-scream clips sourced from public repositories and crowd-recorded samples.
- **Dataset C**: ESC-50 – Environmental Sound Classification dataset with 50 categories including human sounds.

These datasets provided a diverse range of inputs, helping the model generalize to different recording qualities and background noises.

4.3 Metrics for Evaluation

The following metrics were used to evaluate system performance:

- **Accuracy**: Percentage of correctly classified audio samples across all test sets.
- **Precision & Recall**: Especially critical to minimize false positives (non-screams identified as screams) and false negatives (missed screams).
- **F1 Score**: Harmonic mean of precision and recall to provide balanced evaluation.

- **Alert Latency**: Time taken from audio upload to email alert dispatch.

4.4 Results

The system was tested using both predefined datasets and real-time user inputs through the Streamlit interface. The model performance metrics are summarized below:

Metric	**Value**
Accuracy	93.5%
Precision	91.2%
Recall	94.8%
F1 Score	93.0%
Alert Latency	~1.8 seconds

The results reflect strong model performance in identifying scream sounds in noisy environments, validating the effectiveness of MFCC-based features combined with SVM classification.

4.5 Observations

- **High Sensitivity**: The model effectively detects scream patterns even in noisy clips, indicating good feature representation from MFCCs.
- **Low false positives:** Background noises such as sirens, crowd chatter, or animal sounds were rarely misclassified as screams.
- **Streamlit Interface:** Enabled smooth testing and integaction but had limitations in audio recording consistency across browsers
- **Email alerts:** Successfully triggered for ture positive scream events, with minimal delay, showcasing practical emergency response integration.

5. Conclusions:

This project demonstrates a functional and efficient scream detection system using audio processing and machine learning. The integration of MFCC feature extraction with a fine-tuned SVM classifier provided robust performance across varied datasets. The system can serve as an assistive alert mechanism in public safety applications, smart surveillance, or personal safety monitoring systems.

6. Future Work:

- **Deep Learning Integration:** Explore convolutional or recurrent neural networks (CNN/RNN) for improved generalization across unseen environments.
- **Multilingual Voice and Emotion Detection:** Extend capabilities to identify distress in various languages or through emotional cues.
- **Real-Time Streaming Integration:** Implement continuous audio monitoring with low-latency inference using platforms like TensorFlow Lite or ONNX.
- **Mobile Deployment:** Package the system into a lightweight Android/iOS app with push notifications.
- **Edge AI Optimization:** Reduce model size and processing overhead for deployment on low-power devices (e.g., Raspberry Pi, Jetson Nano).

REFERENCES:

[1] Robust unsupervised detection of human screams in noisy acoustic environments" By Mahesh Kumar Nandwana. & Ali Ziaei.. (2015). https://ieeexplore.ieee.org/document/7177952

[2] Application of abnormal sound recognition system for indoor environment" By Chuan-Yu Chang, A., & Yi-Ping Chang, S. (2013). https://ieeexplore.ieee.org/document/6782772

[3] Smart Speaker: Suspicious Event Detection with Reverse Mode Speakers" By Gyorgy Kalmar. (2019). https://ieeexplore.ieee.org/document/8769024

[4] Scream and gunshot detection and localization for audio-surveillance systems" By Valenzise, G., & Gerosa, L. (2007). https://ieeexplore.ieee.org/document/4425280

[5] Excitation Source Feature for Discriminating Shouted and Normal Speech" By Shikha Baghel., & Mahadeva Prasanna, S. (2018). https://ieeexplore.ieee.org/document/8724482

[6] Verbal aggression detection in complex social environments" By Van Hengei, P.W.J., & Andringa, T.c., (2007). https://ieeexplore.ieee.org/document/4425279

[7] Audio Based Violent Scene Detection Using Extreme Learning Machine Algorithm" By Mrunali Mahalle, D., & Dinesh Rojatkar, S. (2021). https://ieeexplore.ieee.org/document/9418209

[8] Detection of audio-based emergency situations using perception sensor network" By Quang Nguyen, & Sang-Seok Yun. (2016). https://ieeexplore.ieee.org/document/7734109

[9] Deep neural networks for automatic detection of screams and shouted speech in subway trains" By Pierre Laffitte, & David Sodoyer. (2016). https://ieeexplore.ieee.org/document/7472921

[10] Two-Stage Supervised Learning-Based Method to Detect Screams and Cries in Urban Environments" By Anil Sharma, & Sanjit Kaul. (2019). https://ieeexplore.ieee.org/document/7348673/

[11] SCREAM: screen-based navigation in voice messages" By Lie, H.W., & Dybvik, P.E,. (2019). https://ieeexplore.ieee.org/document/269551

[12] Special Sound Detection for emergency phones" By Hansheng Lei, & Oscar Valdez. (2013). https://ieeexplore.ieee.org/document/6816306

[13] Sound-event partitioning and feature normalization for robust sound-event detection**"** By Baiying Lei, & Man-Wai Mak. (2014). https://ieeexplore.ieee.org/document/6900692

[14] Sound texture modelling with linear prediction in both time and frequency domains "By Athineos, M., & Ellis, D.P.W (2003). https://ieeexplore.ieee.org/document/1200054