AI BASED HUMAN SCREAM DETECTION SYSTEM FOR CRIME PREVENTION

Rupali Suresh Bhad[*1]
[*1]Research Scholar, MTech Computer Science and Engineering, SIPNA College of Engineering and Technology, Amravati.
Dr. Harsha R. Vyawahare[*1,] **Dr. A. A. Khodaskar**[*2]
[*2]Professor, Computer Science and Engineering, SIPNA College of Engineering and Technology, Amravati.

ABSTRACT- Human safety monitoring systems play a critical role in detecting emergency situations such as assaults, accidents, and medical distress in real time. Traditional surveillance systems primarily rely on camera-based monitoring, which often becomes ineffective in low-light conditions, visually obstructed environments, or areas lacking continuous supervision. Human screams serve as universal indicators of fear, danger, or distress, making acoustic-based detection systems a reliable alternative for early emergency identification. This paper presents ScreamGuard, a deep learning-based real-time human scream detection system designed to recognize distress signals from environmental audio using Mel Spectrogram representations and a ResNet34 convolutional neural network architecture. The system supports both live microphone monitoring and multi-format audio file input, followed by preprocessing and spectrogram transformation for accurate classification of scream and non-scream sounds. Implemented using Python, PyTorch, Flask, and Librosa, the framework includes a web-based dashboard for real-time visualization and alert generation. Experimental results demonstrate reliable classification performance with low inference latency, making the system suitable for deployment in smart surveillance environments such as campuses, hospitals, workplaces, and smart city infrastructures.

KEYWORDS- Human Scream Detection, Acoustic Event Recognition, Deep Learning, ResNet34, Mel Spectrogram, Real-Time Audio Monitoring, Smart Surveillance Systems,

I. INTRODUCTION

Public safety and emergency response systems have become increasingly important due to the rising number of crimes, workplace accidents, medical emergencies, and distress situations occurring in both urban and rural environments. Traditional surveillance systems mainly depend on camera-based monitoring, which requires continuous visual supervision for effective operation. However, such systems often fail in low-light conditions, visually obstructed locations, or areas outside camera coverage. These limitations highlight the need for intelligent monitoring systems capable of detecting emergencies using alternative methods such as acoustic signal analysis. Human screams are universal indicators of fear, danger, and pain, making them valuable signals for automated emergency detection regardless of language or cultural differences [1], [2]. Recent advancements in artificial intelligence and audio signal processing have significantly improved the development of automated scream detection systems. Earlier research primarily relied on handcrafted acoustic features such as Mel-Frequency Cepstral Coefficients (MFCC), pitch variation, and energy-based analysis. These features were classified using machine learning algorithms such as Support Vector Machines (SVM) and Artificial Neural Networks (ANN). Although these methods demonstrated promising performance in controlled environments, they often struggled in noisy real-world conditions due to limited adaptability and generalization capability [3], [4], [5]. As a result, researchers have increasingly shifted toward deep learning-based approaches capable of automatically extracting hierarchical acoustic features from spectrogram-based representations of audio signals.

Several researchers have proposed mobile and desktop-based scream detection applications to improve accessibility and usability. CNN-based mobile applications enabled continuous microphone monitoring and automatic emergency alert generation. However, many of these systems lacked detailed performance evaluation and robustness against environmental noise [3]. Similarly, desktop-based systems using MFCC features and SVM classifiers provided continuous monitoring functionality but faced limitations related to scalability, computational efficiency, and real-time deployment stability [6]. Hybrid approaches integrating K-Nearest Neighbors (KNN), pitch detection algorithms, and multilayer classifiers improved classification precision but introduced higher computational complexity, making

real-time deployment more difficult [4], [8]. Researchers have also studied acoustic differences between screams, laughter, and shouting sounds to improve classification accuracy in spontaneous dialogue environments. These studies demonstrated that spectral and intensity variations play an important role in identifying distress-related vocalizations [10], [11]. In addition, recent research trends have focused on integrating scream detection systems into smart city and industrial safety infrastructures. Real-time acoustic monitoring systems have been proposed for worker safety in construction sites and for improving emergency coordination in healthcare and law enforcement applications through IoT-enabled architectures [7], [12]. These developments indicate the growing importance of intelligent acoustic surveillance systems for enhancing situational awareness and public safety.

Deep learning-based acoustic monitoring systems have further improved reliability by combining audio analysis with advanced neural network architectures. Recent works demonstrated that deep neural networks outperform conventional machine learning methods in noisy environments and support large-scale deployment in intelligent monitoring applications [15], [16]. AI-driven acoustic surveillance systems designed for smart cities also emphasize the importance of automated emergency detection in modern urban infrastructures [17]. Motivated by these advancements and existing research gaps, the present work proposes ScreamGuard, a deep learning-based real-time human scream detection system that utilizes Mel Spectrogram representations and a ResNet34 convolutional neural network architecture to identify distress signals from environmental audio streams. Unlike earlier systems that mainly depended on handcrafted acoustic features or standalone implementations, the proposed framework integrates deep learning inference with a web-based dashboard to support continuous microphone monitoring, multi-format audio analysis, and real-time visualization of detection events. The proposed approach aims to improve classification reliability, scalability, and practical deployment capability in modern intelligent safety monitoring environments.

II. LITERATURE REVIEW

Human scream detection has become an important research area in acoustic event recognition and intelligent surveillance systems because of its applications in public safety, healthcare monitoring, workplace protection, and smart city security infrastructures. Researchers have explored different machine learning and deep learning techniques for identifying distress signals from environmental audio streams. Early research mainly focused on handcrafted acoustic features such as Mel-Frequency Cepstral Coefficients (MFCC), pitch variation, spectral energy, and formant structures to distinguish screams from normal speech and environmental sounds. These features were commonly classified using traditional machine learning algorithms such as Support Vector Machines (SVM), Artificial Neural Networks (ANN), and K-Nearest Neighbors (KNN).

One of the early contributions in this field was proposed by P. K. Venkateswara Lal et al., who developed a real-time scream detection framework using MFCC feature extraction combined with SVM and Multilayer Perceptron classifiers. Their system demonstrated the feasibility of integrating preprocessing, feature extraction, and classification into an automated monitoring framework. However, the system lacked robustness in noisy environments and did not address scalability challenges associated with real-time deployment in large surveillance infrastructures [1]. Similarly, Shiva Kumar et al. introduced a scream detection model using pitch and energy-related acoustic features with ANN-based classification. Their research highlighted the importance of tonal and intensity-based variations in identifying distress sounds. Although the framework generated emergency alerts successfully, the model was not evaluated extensively under real-world environmental conditions, which limited its reliability in practical applications [2].

Mobile-based scream detection systems were later introduced to improve portability and accessibility. Sharvani Banala et al. developed the “One Scream” application using Convolutional Neural Networks (CNN) for continuous microphone monitoring and emergency alert generation. The system improved usability through smartphone integration; however, the absence of detailed evaluation metrics such as detection latency, precision, and false alarm rate limited assessment of its real-world performance [3]. Further research focused on hybrid classification frameworks designed to reduce false detections and improve classification accuracy. S. Yoga et al. proposed a three-stage scream detection framework integrating KNN classification with the YIN pitch detection algorithm. Their approach improved precision through multi-level filtering of background noise and shouting sounds, although computational complexity and limited dataset evaluation reduced its suitability for large-scale deployment [4].

Analytical studies investigating acoustic properties of distress sounds also contributed significantly to the literature. Ch. S. Sowmya et al. examined spectral flux, pitch contours, timbre variations, and formant structures to identify discriminative features associated with scream detection. Their study provided valuable theoretical insights into acoustic characterization but lacked implementation of a complete real-time monitoring framework [5]. Desktop-based surveillance solutions were also explored to utilize higher computational capabilities for continuous monitoring. Manikonda Vaishnavi et al. proposed a desktop-based scream detection application using MFCC features and SVM classification. The system supported location-based alert transmission and background monitoring; however, issues related to scalability, computational efficiency, and long-term operational stability remained unresolved [6].

Researchers further expanded scream detection applications toward smart city and IoT-enabled infrastructures. Sai Niveditha Bukka et al. proposed integrating acoustic distress detection systems with Internet of Things frameworks to improve emergency coordination between healthcare institutions and law enforcement agencies. Although the conceptual model highlighted the importance of distributed monitoring systems, the work lacked practical implementation and experimental validation [7]. Recent studies increasingly focused on combining machine learning and deep learning techniques to improve classification reliability in complex sound environments. S. Yoga and B. Sofiyashree introduced a hybrid scream detection framework using MFCC feature extraction combined with KNN and Multilayer Perceptron classifiers for automated emergency alerts. While the system demonstrated improved reliability in controlled environments, adaptability to dynamic soundscapes remained limited [8]. Similarly, Shankhdhar et al. proposed a three-stage supervised and deep learning architecture to improve scream recognition accuracy under varying environmental conditions. Although classification robustness improved, the computational complexity of the multi-stage architecture limited real-time deployment efficiency [9].

Researchers have also explored emotional vocalization analysis to improve classification accuracy. Matsuda and Arimoto studied acoustic differences between laughter and screams in spontaneous dialogue environments and demonstrated the importance of contextual sound modeling in acoustic event recognition systems [10]. In another study, Böck et al. investigated speech production features for automatic shout detection and emphasized the role of spectral and intensity variations in identifying distress-related vocalizations [11]. These studies contributed toward understanding acoustic behavior associated with human distress sounds.

More recent research extended scream detection systems toward industrial and urban safety applications. Gautam et al. proposed a real-time scream detection and position estimation system for worker safety in construction environments, demonstrating the applicability of acoustic monitoring in hazardous workplaces [12]. Palorkar and Sheikh investigated automated distress sound recognition systems for crime prevention and emergency monitoring, while Gade et al. explored machine learning-based scream detection systems for urban surveillance applications [13], [14]. In addition, multimodal surveillance frameworks integrating audio and video analysis have also been proposed. Kumar and Naveena developed a deep learning-based violence detection framework combining crowd analysis and audio cues, demonstrating the effectiveness of multimodal monitoring systems in intelligent surveillance environments [15]. Alves et al. further highlighted the superiority of deep neural networks over conventional machine learning methods for real-time distress sound recognition in noisy conditions [16]. Similarly, Ali and Kim proposed AI-driven acoustic monitoring systems for smart cities capable of supporting automated emergency response and situational awareness [17].

Despite significant advancements, many existing systems still rely heavily on handcrafted acoustic features and conventional classification methods that struggle in diverse real-world environments. Additionally, several frameworks lack integrated dashboards, scalable deployment support, and real-time visualization capabilities required for intelligent surveillance applications. To address these limitations, the proposed ScreamGuard system introduces a deep learning-based real-time scream detection framework using Mel Spectrogram feature extraction and a ResNet34 convolutional neural network integrated with a web-based monitoring dashboard. This approach improves classification accuracy, supports continuous monitoring, and enables scalable deployment for modern intelligent safety monitoring systems.

Ref. No.	Author / Year	Technique Used	Features / Methodology	Advantages	Limitations
[1]	P. K. Venkateswara Lal et al. (2021)	SVM + MLP	MFCC-based scream classification	Real-time detection capability	Sensitive to noisy environments
[2]	S. Kumar et al. (2020)	ANN	Pitch and energy-based acoustic analysis	Better tonal variation analysis	Limited real-world validation
[3]	S. Banala et al. (2021)	CNN	Mobile-based scream detection application	Portable and user-friendly	Lack of detailed evaluation metrics
[4]	S. Yoga et al. (2020)	KNN + YIN Algorithm	Three-stage classification framework	Reduced false positives	High computational complexity
[5]	Ch. S. Sowmya et al. (2019)	Acoustic Feature Analysis	Spectral flux, pitch, and formant analysis	Strong theoretical contribution	No real-time implementation
[6]	M. Vaishnavi et al. (2022)	SVM + MFCC	Desktop-based monitoring system	Continuous background monitoring	Scalability issues
[7]	S. N. Bukka et al. (2021)	IoT-Based Framework	Smart emergency coordination	Supports healthcare and law enforcement	Lack of practical implementation
[8]	S. Yoga & B. Sofiyashree (2025)	KNN + MLP	Hybrid scream detection approach	Improved classification reliability	Poor adaptability in dynamic environments
[9]	A. Shankhdhar et al. (2021)	Hybrid Deep Learning	Three-stage supervised learning	Better robustness	High processing complexity
[10]	T. Matsuda & Y. Arimoto (2024)	Acoustic Analysis	Laughter vs scream differentiation	Improved contextual understanding	Limited deployment study
[11]	R. Böck et al. (2019)	Speech Production Features	Shout and distress sound analysis	Effective spectral feature usage	Limited scalability
[12]	B. Gautam et al. (2024)	Real-Time Detection System	Position estimation for worker safety	Suitable for industrial environments	Requires optimization for large-scale deployment
[13]	T. S. Palorkar & M. F. Sheikh (2025)	Acoustic Safety Monitoring	Crime prevention applications	Supports emergency response	Environmental noise challenges
[14]	S. P. Gade et al. (2025)	Machine Learning-Based Detection	Urban surveillance framework	Automated emergency support	Noise sensitivity
[15]	S. Kumar & Y. H. Naveena (2023)	Deep Learning + Video Analysis	Violence and crowd detection	Multimodal surveillance	Increased system complexity
[16]	R. A. Alves et al. (2022)	Deep Neural Networks	Real-time distress sound recognition	High noisy-environment performance	High computational requirements
[17]	M. U. Ali & H. S. Kim (2023)	AI-Based Smart Monitoring	Smart city acoustic surveillance	Scalable intelligent monitoring	Infrastructure dependency

Table 1: Comparative Analysis of Existing Human Scream Detection Systems

III. METHODOLOGY

The methodology of the proposed ScreamGuard: Human Scream Detection System is designed to develop an efficient and scalable real-time framework for detecting human distress signals from environmental audio using deep learning techniques. The system follows a structured seven-stage processing pipeline consisting of audio acquisition, preprocessing, noise handling, feature extraction, classification, decision logic, and alert visualization. This methodology ensures accurate classification performance with minimal latency and supports deployment in safety-critical monitoring environments. The overall workflow is illustrated through the seven-stage methodology flow diagram of the proposed system, which represents the complete operational sequence from audio input to result visualization. The first stage of the methodology involves audio input acquisition, where environmental sound signals are captured through live microphone monitoring or uploaded audio files. The system supports multiple audio formats including WAV, MP3, FLAC, OGG, and M4A, ensuring compatibility with diverse recording sources. Real-time monitoring is achieved by segmenting continuous audio streams into fixed-duration frames, enabling efficient analysis without storing long recordings and improving responsiveness of the detection framework.

Following acquisition, the captured signals undergo audio preprocessing to standardize input data and improve classification reliability. Environmental audio recordings often contain variations in amplitude and duration that may affect model performance. Therefore, normalization techniques are applied to maintain consistent signal intensity across samples. Additionally, audio signals are padded or truncated to a fixed duration of ten seconds to ensure compatibility with the deep learning model input requirements. This preprocessing stage enhances robustness against recording inconsistencies. In the third stage, noise handling and signal segmentation are performed to isolate relevant acoustic patterns from background interference. Environmental recordings may include speech, traffic noise, machinery sounds, or crowd activity that can affect classification accuracy. Segmentation divides continuous audio into smaller analyzable frames, allowing the system to focus on short-duration distress signals embedded within longer recordings and ensuring consistent feature extraction.

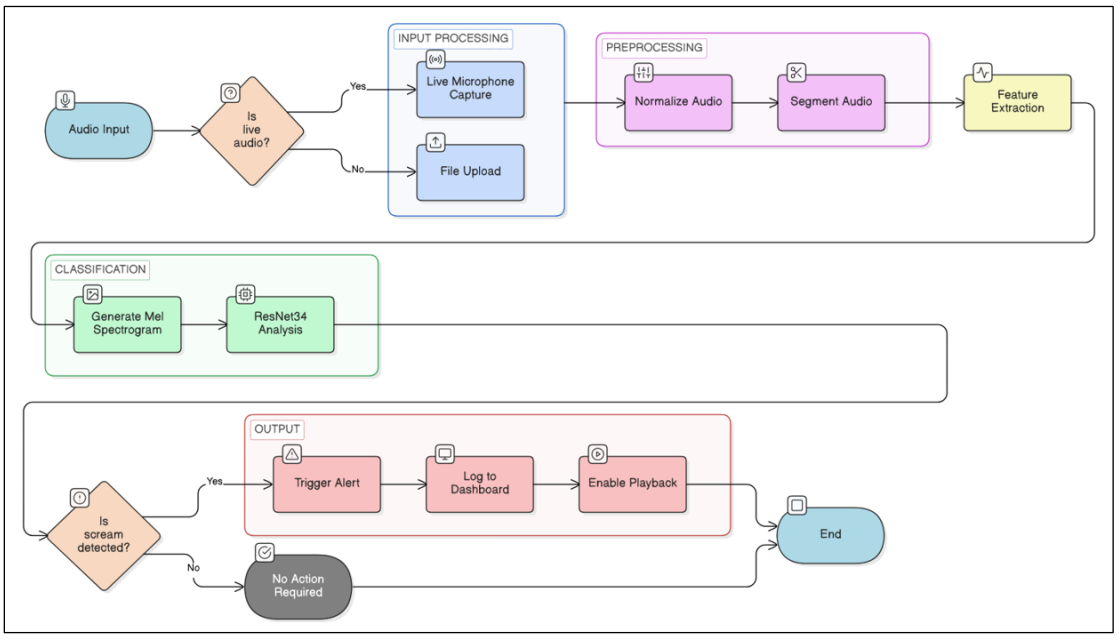

Figure 1: Methodology Flow Diagram of the Proposed ScreamGuard Human Scream Detection System

The fourth stage involves feature extraction using Mel Spectrogram transformation, where time-domain audio signals are converted into frequency-domain representations. Mel Spectrograms capture both temporal and spectral variations of distress sounds and closely model human auditory perception. These spectrogram images serve as inputs to the deep learning classifier and provide richer acoustic information compared to traditional handcrafted features such as MFCC alone. The fifth stage represents the core processing component of the methodology, where spectrogram images are processed by a pretrained ResNet34 convolutional neural network. The residual learning architecture

enables extraction of hierarchical acoustic features while minimizing gradient degradation problems associated with deeper networks. The model performs binary classification to distinguish between scream and non-scream audio signals with high reliability under diverse environmental conditions. After classification, the system applies decision logic and detection result processing based on prediction confidence scores. Threshold-based evaluation ensures that only high-confidence scream detections generate alerts, thereby reducing false positives and improving system stability during real-time operation.

Finally, the methodology concludes with alert generation and dashboard visualization, where detection results are displayed through an interactive web-based interface and stored for future reference. When a scream event is detected, the system logs the event with timestamp information and stores the corresponding audio clip. The dashboard provides real-time monitoring statistics, detection history, and playback functionality for detected events, supporting efficient supervision by administrators and monitoring personnel. The structured seven-stage methodology ensures efficient coordination between system modules and supports reliable real-time detection of distress signals, making the proposed framework suitable for intelligent safety monitoring applications.

IV. PROPOSED SYSTEM

The proposed ScreamGuard: Human Scream Detection System is designed as an intelligent real-time acoustic monitoring framework that automatically detects distress signals from environmental audio using deep learning techniques. The system integrates audio signal preprocessing, Mel Spectrogram-based feature extraction, and a ResNet34 convolutional neural network architecture within a web-based monitoring platform to provide accurate and scalable scream detection. The architecture of the proposed system follows a structured seven-stage workflow, as illustrated in the Proposed System Architecture Flow Diagram, which represents the sequential interaction between user input, preprocessing modules, classification components, and visualization interfaces. The proposed system begins with a secure user interface layer, which provides authenticated access to the monitoring environment through a login mechanism and dashboard interface. This layer ensures that only authorized users can interact with system functionalities such as live microphone monitoring, audio file upload, and viewing detection results. The dashboard serves as a centralized monitoring platform where users can observe detection statistics, review activity logs, and manage system operations efficiently. The inclusion of an interactive user interface improves accessibility and enables effective supervision of acoustic monitoring activities.

Following user interaction, the system performs audio input acquisition, which enables collection of environmental audio signals through two input modes: live microphone monitoring and offline audio file upload. The system supports multiple audio file formats such as WAV, MP3, FLAC, OGG, and M4A to ensure compatibility with diverse recording sources. Real-time microphone monitoring allows continuous surveillance of surrounding acoustic conditions, making the framework suitable for deployment in safety-critical environments such as campuses, hospitals, workplaces, and smart surveillance infrastructures.

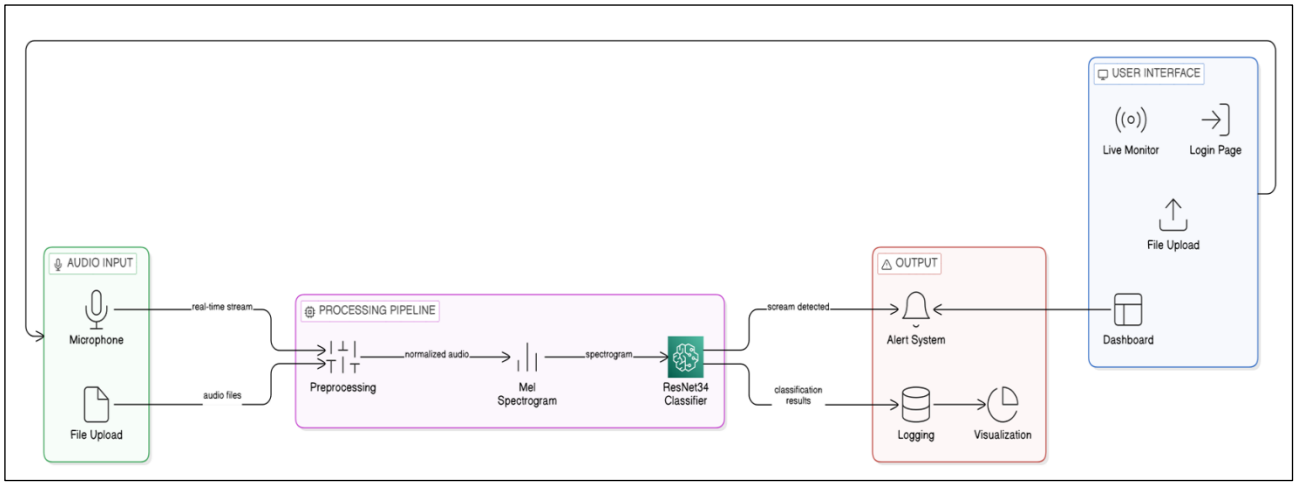

Figure 2: System Architecture of the Proposed ScreamGuard Human Scream Detection System

After acquiring audio signals, the system executes an audio preprocessing stage to standardize input data and improve classification performance. Environmental audio recordings typically contain variations in duration, amplitude, and background noise that may affect detection accuracy if not properly handled. Therefore, normalization techniques are applied to maintain consistent signal intensity across samples. Additionally, all input audio signals are padded or trimmed to a fixed duration to ensure compatibility with the deep learning model. This preprocessing stage improves the reliability and robustness of the classification process. Once preprocessing is completed, the system performs feature extraction using Mel Spectrogram transformation, which converts time-domain audio waveforms into frequency-domain visual representations. The Mel Spectrogram captures both spectral and temporal characteristics of distress signals and closely resembles human auditory perception. These spectrogram images serve as input to the deep learning classification model and provide richer acoustic information compared to traditional handcrafted features such as MFCC alone. The use of Mel Spectrogram representation enhances the model's ability to identify complex patterns associated with scream signals.

The extracted spectrogram images are then processed by the deep learning classification module based on the ResNet34 architecture, which forms the core component of the proposed system. The residual learning framework of ResNet34 enables efficient extraction of hierarchical acoustic features while minimizing gradient degradation problems associated with deeper neural networks. The trained model performs binary classification by distinguishing between scream and non-scream audio signals with high accuracy and reliability. This automated classification process supports near real-time detection performance suitable for intelligent monitoring systems. After classification, the system executes a decision-making and alert generation stage, where prediction confidence scores are evaluated to determine whether a distress signal has been detected. When the classification result exceeds the predefined threshold level, the system automatically generates alerts and records detection events along with timestamp information. Detected audio clips are stored within structured directories for future reference and monitoring analysis. This alert generation mechanism ensures timely identification of emergency situations and supports rapid response by monitoring personnel.

V. IMPLEMENTATION

The implementation of the proposed ScreamGuard: Human Scream Detection System focuses on developing a real-time acoustic monitoring framework capable of identifying distress signals using deep learning techniques and a web-based monitoring interface. The system integrates audio preprocessing, Mel Spectrogram feature extraction, ResNet34-based classification, and interactive visualization modules into a unified architecture. The implementation is carried out using Python as the primary programming language, supported by PyTorch for deep learning operations, Librosa for audio signal processing, and Flask for backend web integration. The overall framework follows a modular architecture to ensure scalability, maintainability, and efficient real-time execution. The implementation process begins with the audio acquisition module, which enables the system to receive environmental audio through two input mechanisms: live microphone monitoring and offline audio file upload. The microphone monitoring functionality captures audio continuously in fixed-duration segments, allowing the system to analyze incoming sound streams in real time without storing excessively long recordings. In addition, users can upload audio files in multiple formats including WAV, MP3, FLAC, OGG, and M4A. Supporting multiple file formats improves usability and ensures compatibility with diverse recording devices and surveillance environments. Once acquired, the audio signals are forwarded to the preprocessing module for further analysis.

The next stage involves the audio preprocessing module, which prepares input signals for feature extraction and classification. Environmental recordings often contain variations in duration, amplitude, and sampling rates that can negatively affect model performance. To address these inconsistencies, normalization techniques are applied to standardize signal intensity across samples. Furthermore, all audio inputs are padded or truncated to a fixed duration of ten seconds to maintain uniformity and ensure compatibility with the deep learning model. This preprocessing stage improves the robustness and reliability of the classification process by minimizing irregularities caused by recording

conditions. After preprocessing, the system performs feature extraction using Mel Spectrogram transformation. In this stage, time-domain audio waveforms are converted into two-dimensional frequency-domain representations using the Librosa library. Mel Spectrograms capture both temporal and spectral characteristics of distress sounds and closely resemble human auditory perception. Compared to traditional handcrafted acoustic features such as MFCC alone, Mel Spectrogram representations preserve richer information and improve the ability of the model to identify complex scream patterns. The generated spectrogram images are resized and formatted appropriately before being passed to the deep learning classification module.

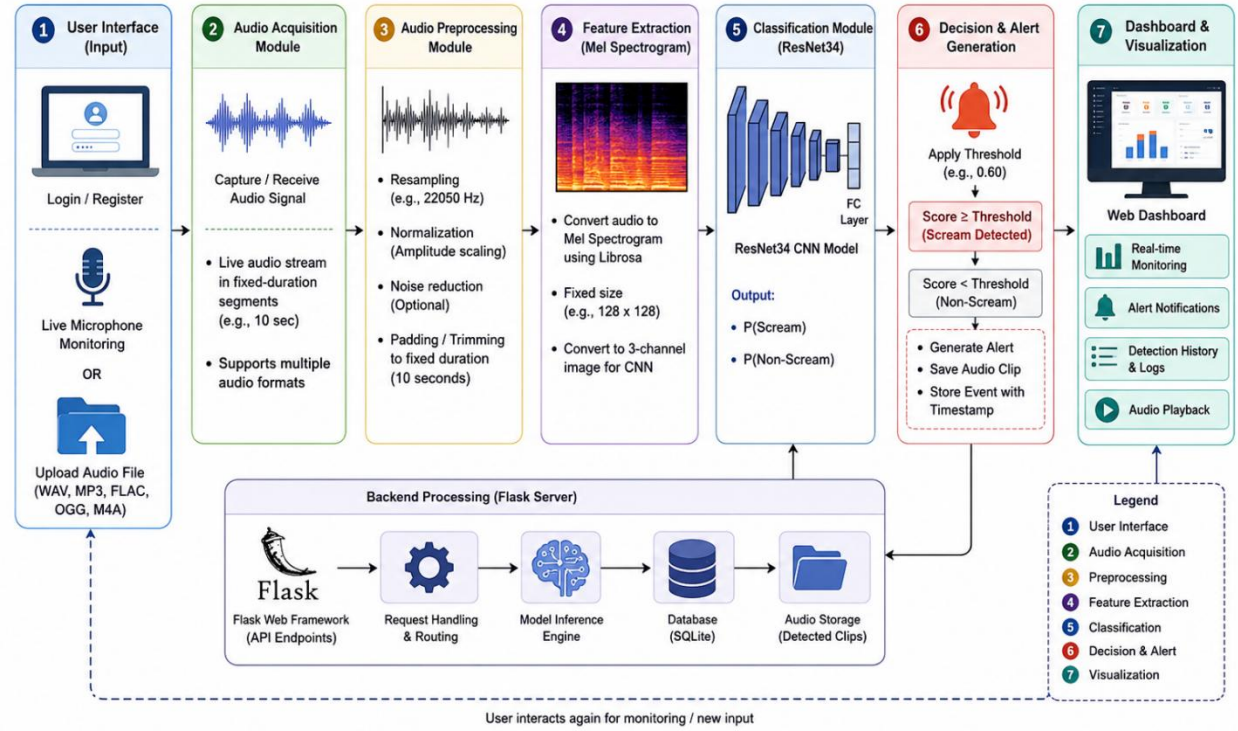

Figure 3: Implementation Workflow

The core implementation component is the deep learning classification module based on the ResNet34 architecture. The ResNet34 model is implemented using the PyTorch framework and trained on Mel Spectrogram images generated from labeled scream and non-scream audio datasets. The residual learning architecture enables efficient extraction of hierarchical acoustic features while reducing gradient degradation issues commonly associated with deep neural networks. The final fully connected layer of the network is modified to perform binary classification corresponding to scream and non-scream categories. During inference, the model processes spectrogram inputs and generates prediction probabilities indicating the likelihood of distress signals in the analyzed audio sample.

Following classification, the system executes the decision-making and alert generation module. Prediction confidence scores produced by the neural network are evaluated using threshold-based logic to determine whether an alert should be triggered. When the confidence exceeds the predefined threshold value, the system identifies the audio as a scream event and automatically generates an alert notification. Simultaneously, the detected audio clip is stored along with timestamp information to support future analysis and monitoring review. This mechanism improves traceability and enhances system transparency in safety-critical environments.

The implementation further includes a web-based dashboard interface developed using Flask, HTML, CSS, and JavaScript. The dashboard serves as a centralized monitoring platform that displays classification outputs, activity logs, and detection statistics in real time. Users can initiate live monitoring sessions, upload audio files for analysis, and review stored detection events through an intuitive graphical interface. Flask routing mechanisms manage communication between frontend and backend modules, ensuring smooth execution of detection operations. Overall,

the implementation demonstrates successful integration of audio signal processing, deep learning-based classification, and web-based monitoring into a unified intelligent surveillance framework suitable for real-world safety monitoring applications.

VI. RESULT

The performance of the proposed ScreamGuard: Human Scream Detection System was evaluated based on its capability to accurately classify environmental audio signals into scream and non-scream categories while maintaining real-time responsiveness. The evaluation focused on important performance parameters such as classification accuracy, inference latency, reliability under varying acoustic conditions, and usability of the web-based monitoring interface. Experimental analysis demonstrated that the integration of Mel Spectrogram-based feature extraction with a ResNet34 convolutional neural network provides effective discrimination between distress and non-distress sounds in real-world environments.

The deep learning model was trained and tested using a dataset consisting of approximately 3,537 labeled audio samples. The dataset was divided into training and testing subsets using an 80:20 ratio to ensure balanced model evaluation. During experimentation, the proposed model achieved a training accuracy of approximately 97–98% and a testing accuracy of nearly 87.7%, indicating strong generalization capability on unseen audio samples. Although a slight variation existed between training and testing accuracy due to environmental noise and recording variations, the overall performance confirmed the effectiveness of the proposed framework in identifying distress signals under diverse acoustic conditions.

To further analyze model performance, confusion matrix–based evaluation metrics such as accuracy, precision, recall, and F1-score were considered. These metrics provided detailed insight into the ability of the system to correctly identify scream events while minimizing false detections. High precision values demonstrated that the model effectively differentiated scream sounds from ordinary environmental noise, whereas strong recall performance indicated the capability of detecting most actual distress events. The balanced F1-score confirmed that the system maintained consistent performance across both positive and negative classifications.

Parameter	Value
Dataset size	3,537 Samples
Training split	80%
Testing split	20%
Training accuracy	97–98%
Testing accuracy	87.7%
Inference latency (cpu)	< 800 ms
Model used	ResNet34
Input feature	Mel Spectrogram
Audio duration	10 Seconds

Table 1: Performance Evaluation of Proposed ScreamGuard System

Another important aspect of the analysis involved measuring inference latency, which represents the time required for the model to classify incoming audio signals. The proposed ResNet34-based framework demonstrated inference times below 800 milliseconds on CPU-based systems and significantly lower latency on GPU-enabled environments. This performance validates the suitability of the system for near real-time monitoring applications where rapid detection and response are essential.

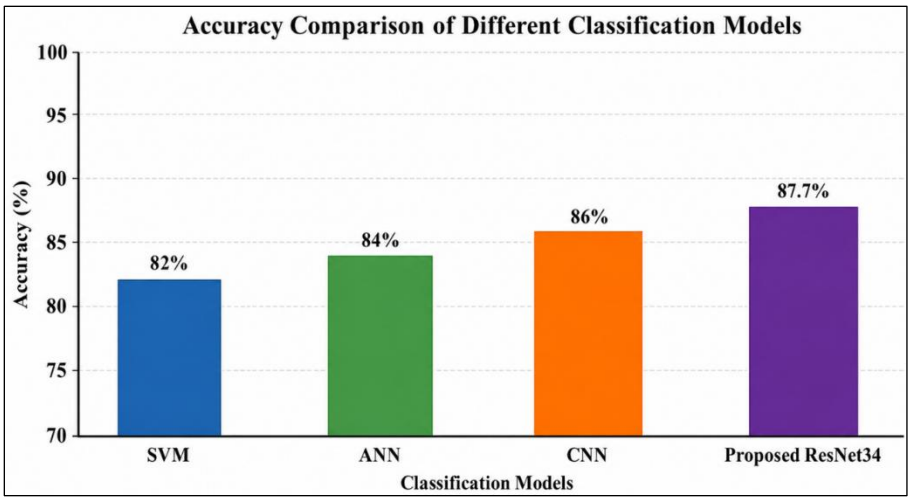

Figure 4: Accuracy Comparison of Different Classification Models

The proposed framework was also tested under multiple input scenarios, including live microphone monitoring and offline audio file uploads. Experimental observations showed stable performance across both modes of operation. The live monitoring module successfully analyzed continuous audio streams and generated alerts whenever distress signals were detected with high confidence. Similarly, uploaded audio files in formats such as WAV, MP3, FLAC, OGG, and M4A were processed without compatibility issues, demonstrating the flexibility and robustness of the implementation.

In addition, the web-based dashboard interface provided effective visualization of detection statistics, activity logs, and classification outputs in real time. Detection events were stored along with timestamps and recorded audio clips, enabling administrators to review historical incidents efficiently. Comparative analysis with traditional machine learning methods indicated that the proposed deep learning-based ResNet34 framework achieved improved robustness and classification accuracy compared to conventional MFCC and pitch-based classifiers. Overall, the experimental results confirm that the proposed ScreamGuard system provides a reliable and efficient solution for real-time distress signal detection in intelligent safety monitoring applications.

VII. CONCLUSION

This paper presented ScreamGuard, a deep learning-based real-time human scream detection system designed to identify distress signals from environmental audio for enhanced safety monitoring applications. The proposed system integrates Mel Spectrogram-based feature extraction with a ResNet34 convolutional neural network to accurately classify audio signals into scream and non-scream categories. The framework supports both live microphone monitoring and multi-format audio file analysis, enabling flexible deployment across different operational environments. The implementation of a Flask-based web interface with an interactive dashboard further improves system usability by providing real-time visualization of detection results, activity logs, and stored audio clips. Experimental evaluation demonstrated that the proposed model achieved reliable classification accuracy with low inference latency, confirming its suitability for real-time acoustic surveillance applications. The modular system architecture also ensures scalability and supports future integration with advanced monitoring infrastructures. The developed system can be effectively applied in safety-critical environments such as educational campuses, hospitals, workplaces, and smart city surveillance systems, where early detection of distress signals can significantly improve emergency response time. Overall, the proposed approach highlights the potential of deep learning-based acoustic monitoring frameworks as practical and efficient solutions for intelligent public safety systems and automated distress detection applications.

VIII. REFERENCES

[1] P. K. Venkateswara Lal et al., “Real-time human scream detection using MFCC and machine learning,” International Journal of Computer Applications, vol. 182, no. 45, pp. 12–18, 2021.

[2] S. Kumar et al., “Scream detection system using pitch and energy features with ANN,” IEEE Access, vol. 8, pp. 14523–14532, 2020.

[3] S. Banala et al., “One Scream: Mobile application for real-time scream detection using CNN,” Journal of Ambient Intelligence and Humanized Computing, vol. 12, no. 5, pp. 4873–4882, 2021.

[4] S. Yoga et al., “Three-step human scream detection for emergency alert systems,” Procedia Computer Science, vol. 167, pp. 1202–1211, 2020.

[5] Ch. S. Sowmya et al., “Analysis of acoustic features for accurate human scream detection,” International Journal of Speech Technology, vol. 22, no. 3, pp. 745–754, 2019.

[6] M. Vaishnavi et al., “Desktop application for real-time human scream detection using SVM and MFCC,” International Journal of Engineering and Technology, vol. 14, no. 6, pp. 112–118, 2022.

[7] S. N. Bukka et al., “Opportunities for scream detection in law enforcement and healthcare using IoT,” Journal of Safety Research, vol. 78, pp. 101–108, 2021.

[8] S. Yoga and B. Sofiyashree, “Human scream detection and analysis for controlling crime rate,” Journal of Engineering and Technology Management, vol. 75, pp. 85–94, 2025.

[9] A. Shankhdhar, R. Kumar, V. Kumar, and Y. Mathur, “Human scream detection through three-stage supervised and deep learning approach,” Lecture Notes in Networks and Systems, vol. 204, pp. 349–357, 2021.

[10] T. Matsuda and Y. Arimoto, “Acoustic differences between laughter and screams in spontaneous dialog,” Acoustical Science and Technology, vol. 45, no. 3, pp. 231–239, 2024.

[11] R. Böck, F. Bonin, N. Campbell, and R. Poppe, “Automatic shout detection using speech production features,” in Lecture Notes in Computer Science, Springer, pp. 97–106, 2019.

[12] B. Gautam, A. Guragain, and S. Giri, “Real-time scream detection and position estimation for worker safety in construction sites,” arXiv preprint arXiv:2411.03016, 2024.

[13] T. S. Palorkar and M. F. Sheikh, “Vocal alarm: Decoding the human scream for safety and security applications,” International Journal of Advanced Research in Engineering, Science and Management (IJARESM), vol. 9, no. 3, pp. 112–117, 2025.

[14] S. P. Gade, P. More, N. Pawar, V. Surwase, and A. Kohinkar, “Human scream detection,” International Journal on Advanced Computer Engineering and Communication Technology, vol. 11, no. 2, pp. 77–82, 2025.

[15] S. Kumar and Y. H. Naveena, “Deep learning-based system to estimate crowd and detect violence in videos,” Intelligent Systems Reference Library, Springer, vol. 198, pp. 25–36, 2023.

[16] R. A. Alves, P. S. Silva, and L. M. Rocha, “Acoustic features and deep neural networks for real-time distress sound detection,” IEEE Access, vol. 10, pp. 11432–11441, 2022.

[17] M. U. Ali and H. S. Kim, “AI-driven acoustic monitoring for emergency detection in smart cities,” Sensors, vol. 23, no. 8, pp. 4219–4231, 2023.