A Comparative Analysis of CNN and CRNN Models for Home Emergency Sound Detection

[1]AML F. ELLAFI ,[2]ZAHRA A.MATRAW,[3] MINATH ALLAH E.ALSHEGWEE

[1,2,3]Department of Computer Engineering, College of Electronic Technology, Bani Walid, Libya

[1]amlellafi92@gmail.com [2] zahramatraw@gmail.com [3]minathallahebraheemalamin@gmail.com

Abstract_The rise in single-person households underscores the critical need for reliable, privacy-preserving home monitoring systems. This paper presents a comprehensive comparative study between a Convolutional Neural Network (CNN) and a Convolutional Recurrent Neural Network (CRNN) for detecting domestic emergency sounds. A robust pipeline was implemented, involving the curation of a balanced dataset of normal and emergency sounds, extensive data augmentation, and feature extraction using Mel-Frequency Cepstral Coefficients (MFCCs). Counter to the theoretical expectation that CRNNs would excel at modeling temporal audio patterns, our experimental results demonstrate the clear superiority of the CNN model. The CNN achieved a remarkable accuracy of 98% and a weighted F1-score of 0.98, outperforming the CRNN (95% accuracy). Furthermore, the CNN exhibited faster convergence, greater training stability, and superior generalization. These findings indicate that for short-duration, spectrally distinct emergency sounds, the spatial feature extraction of CNNs is not only sufficient but more effective than explicit temporal modeling with CRNNs. The study concludes that the CNN architecture is the optimal choice for developing efficient and reliable audio-based emergency detection systems for resource-constrained smart home environments.

Keywords_ Emergency Sound Detection, Deep Learning, Convolutional Neural Network (CNN), Convolutional Recurrent Neural Network (CRNN), Smart Home, Audio Classification.

I. INTRODUCTION

The global rise in single-person households (SPHs), now representing 15-20% of households in developed nations [1], has intensified the need for reliable home safety solutions. This demographic shift creates particular vulnerability during emergencies where immediate assistance may be unavailable. The World Health Organization reports approximately 684,000 annual fatalities from falls alone, with older adults experiencing the highest risk [2].

Traditional monitoring systems face significant limitations. Visual surveillance raises substantial privacy concerns [3], while wearable devices suffer from compliance issues [4]. These challenges have accelerated interest in privacy-preserving audio-based monitoring, leveraging the distinct acoustic signatures produced by emergency events such as cries for help, breaking glass, and fire alarms [5].

Deep learning approaches, particularly Convolutional Neural Networks (CNNs) and Convolutional Recurrent Neural Networks (CRNNs), have demonstrated remarkable success in audio event detection [7]. However, existing commercial systems like Amazon's Alexa Guard and Google's Nest remain limited to narrow sound sets [8], lacking capability for broader emergency detection including screams, falls, and gas leaks. Furthermore, despite CRNNs' theoretical advantages for temporal modeling, empirical comparisons between CNNs and CRNNs in home environments remain scarce [9,10].

This study addresses these gaps through systematic comparison of CNN and CRNN architectures for home emergency sound detection. Our main contributions include: (1) designing a robust audio processing pipeline, (2) conducting comprehensive architectural comparison on a balanced multi-class dataset, and (3) evaluating practical deployment considerations including training stability and computational efficiency.

II. LITERATURE REVIEW

A. Evolution of Home Monitoring Systems

When we traced the historical evolution of home safety systems, we observed that early solutions primarily relied on visual surveillance and wearable sensors. Through our analysis of the literature, we found that camera-based monitoring systems—while effective—face persistent privacy and security challenges [3,11]. Similarly, we noted that wearable devices such as panic buttons and fall detectors often suffer from user compliance issues and battery dependency [4,12].

These limitations we identified in previous work directly led us to explore audio-based monitoring as a privacy-preserving alternative capable of detecting critical acoustic events without recording personal images or conversations [5,13]. However,

DOI:10.5281/zenodo.17553161

when we delved deeper into analyzing these studies, we observed that most focused on a narrow range of sounds (e.g., alarms or breaking glass) and lacked robustness against background noise and overlapping acoustic events. This limited scope we observed in previous literature highlights a critical research gap that our current study aims to address by expanding detection capabilities to include diverse emergency sounds such as screams, falls, and gas leaks.

B. Deep Learning in Audio Classification

In our comprehensive assessment of deep learning literature for audio classification, we observed that the transition from hand-crafted features to learned representations has radically transformed audio event detection. We found that Convolutional Neural Networks (CNNs) demonstrate remarkable effectiveness in extracting local spectral patterns from short audio segments [7,14]. In contrast, our review revealed that Recurrent Neural Networks (RNNs) and their variants (LSTM, GRU) excel at modeling temporal dependencies in longer audio sequences [15,16].

Through our cumulative analysis of the literature, we observed the emergence of several hybrid Convolutional Recurrent Neural Network (CRNN) models that combine the spatial feature extraction capabilities of CNNs with the temporal modeling strengths of RNNs [10,17]. Despite the impressive results achieved by these studies, we found they often overlooked computational efficiency considerations and real-time processing constraints—aspects we consider crucial for practical smart home applications.

Most importantly, we noticed a remarkable scarcity of direct empirical comparisons between CNN and CRNN architectures under identical experimental conditions. This gap in systematic comparison made it challenging to determine which approach is more suitable for detecting short-duration emergency sounds in practical scenarios, which is exactly what we aimed to address in our experimental design.

C. Current Challenges and Research Gaps

Through our critical analysis of previous studies, we were able to identify three main research gaps that require urgent attention:

The first gap we observed lies in the limited scope of sound categories being studied, coupled with a noticeable shortage of comprehensive datasets that accurately represent the acoustic diversity of real-world home environments [8,18]. These data scope limitations we attempted to address directly in the current study by developing a balanced and comprehensive dataset.

The second gap we identified concerns the insufficient comparative studies between Convolutional Neural Networks (CNN) and Convolutional Recurrent Neural Networks (CRNN) models, despite their fundamental differences at both conceptual and application levels [10,19]. This particular gap forms the main focus of our current study, where we designed systematic comparative experiments to address this shortcoming.

The third gap emerges from the limited attention we observed given to models' operational efficiency factors and their practical applicability in edge computing environments [20]. These practical considerations we gave top priority in our experimental design and model evaluation.

This study directly addresses these limitations by providing a systematic comparative analysis of CNN and CRNN architectures using a balanced, multi-class emergency sound dataset. Moreover, it offers practical insights for optimizing model design in real-world smart home environments, with particular emphasis on computational efficiency, training stability, and generalization capability.

II. METHODOLOGY

This section outlines the systematic approach adopted to design, develop, and evaluate the deep learning-based emergency sound detection system. The methodology is structured into five core phases: data collection and preprocessing, feature extraction, model development, training procedure, and evaluation. This structured pipeline ensures reproducibility and a fair comparative analysis between the proposed models.

A. Data Collection and Preprocessing

A composite and balanced dataset was curated from multiple public audio repositories to ensure diversity and representativeness.

Primary sources included **UrbanSound8K**, **ESC-50**, and specialized Kaggle datasets for sounds like screams, falls, and gunshots. This was supplemented with samples from **Freesound** and **Pixabay** to broaden the acoustic variability.

- **Normal Sounds**
  3,000 samples encompassing common domestic noises (e.g., talking, music, appliance hum).
- **Emergency Sounds**
  Ten critical event classes, including baby crying, breaking glass, explosion, falling, fire, gas leak, gunshot, scream, and alarm, with approximately 350-390 samples per class initially.

To address the inherent class imbalance and prevent model bias, a rigorous data augmentation strategy was applied **exclusively** to the emergency sound classes. The

DOI:10.5281/zenodo.17553161

augmentation techniques, implemented using the Librosa library, included:

- **Pitch Shifting**
  Altering the pitch by ±3 semitones.
- **Time Stretching**
  Modifying the tempo by factors between 0.7 and 1.3.
- **Noise Injection**
  Adding low-amplitude white noise to simulate real-world conditions.

This process increased the emergency class samples by a factor of eight, significantly balancing the dataset as detailed in **Table 1**

Table 1: Summary of dataset classes and total audio duration

Sub Category	Original Sample Count	Estimated Duration (Minutes)
Normal	3000	250.0
Baby crying	379	31.6
Breaking	382	31.8
Explosion	360	30.0
Falling	320	26.7
Fire	332	27.7
Gas	350	29.2
Gunshot	340	28.3
Scream	381	31.8
Alarm	350	29.2

All audio clips were standardized to a uniform length of 5 seconds through truncation or zero-padding as illustrated in **Fig 1**. They were converted to a monaural channel and resampled at a consistent rate of 22.05 kHz. Amplitude normalization was applied to minimize variability from different recording conditions.

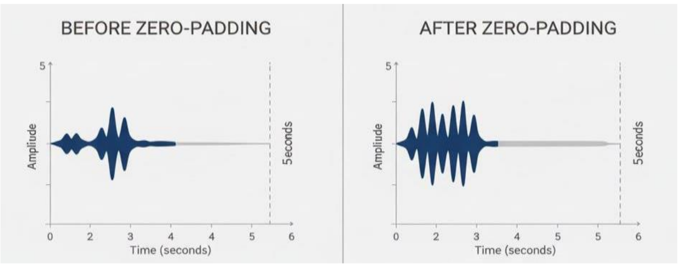


Fig 1: An audio waveform before and after zero-padding

B. Feature Extraction

To transform the raw audio signals into a format suitable for deep learning models, **Mel-Frequency Cepstral Coefficients (MFCCs)** were extracted. MFCCs were selected for their perceptual relevance, as the Mel scale approximates the human auditory system's response, and their proven efficacy in audio classification tasks [11, 6].

The extraction process, performed using Librosa and detailed in **Fig 2**, involved:

1. **Framing**
   Splitting the audio signal into short, overlapping frames.
2. **STFT & Mel Filter bank**
   Applying a Short-Time Fourier Transform (STFT) and mapping the power spectrum to the Mel scale using a filter bank of 40 triangular filters.
3. **Log-compression & DCT**
   Computing the logarithm of the filter bank energies and applying a Discrete Cosine Transform (DCT) to decorrelate the output, resulting in 40 cepstral coefficients.

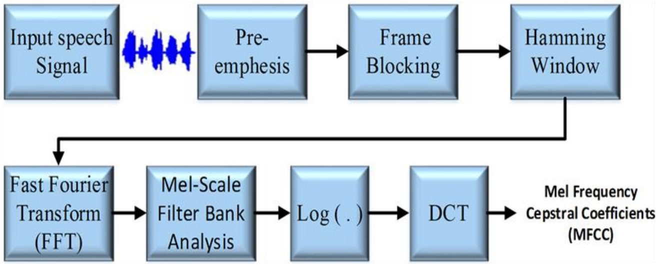

Input speech Signal
Pre-emphasis
Frame Blocking
Hamming Window
Fast Fourier Transform (FFT)
Mel-Scale Filter Bank Analysis
Log (.)
DCT
Mel Frequency Cepstral Coefficients (MFCC)

Fig 2: MFCCs Based Feature Extraction Steps

The resulting MFCCs form a 2D spectro-temporal representation (40 coefficients × time frames). The time axis was standardized to 216 frames, creating a consistent input matrix of dimensions (40, 216, 1) for all samples, which can be treated as a single-channel image.

C. *Model Architectures*

Two distinct deep learning architectures were implemented and compared using the TensorFlow-Keras framework.

1. Convolutional Neural Network (CNN)

The CNN model was designed as a baseline to exploit the spectro-temporal patterns in the MFCC . The architecture illustrated in **Fig 3**.

a. Three Convolutional Blocks

Each block contains two Conv2D layers (with 32, 64, and 128 filters, respectively, using a 3×3 kernel and ReLU activation),

DOI:10.5281/zenodo.17553161

followed by Batch Normalization, MaxPooling2D (2×2), and Dropout (rate=0.25).

b. Classification Head

The feature maps are flattened and passed through two Dense layers (256 and 128 units, ReLU activation), each followed by Batch Normalization and a higher Dropout rate (0.5) for regularization. The final output layer is a Dense layer with a softmax activation function for multi-class classification.

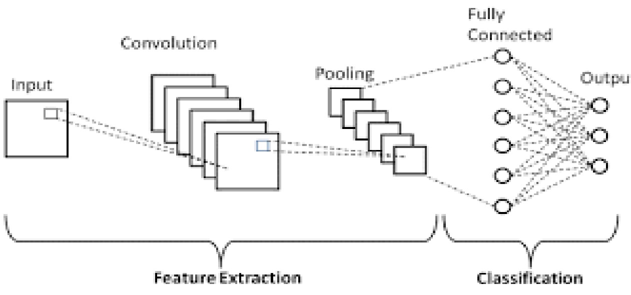


Fig 3 Architectural diagram of the CNN model for audio classification

c. Convolutional Recurrent Neural Network (CRNN)

The CRNN model was developed to capture both spatial and temporal dependencies. It integrates the feature extraction power of CNNs with the sequence modeling capability of RNNs, with its architecture illustrated in **Fig 4**.

a) Feature Extraction Frontend

The model uses the first two convolutional blocks from the CNN model, producing a sequence of high-level feature maps.

b) Temporal Modeling Backend

The feature maps are reshaped into a time sequence and fed into two Gated Recurrent Unit (GRU) layers (128 and 64 units, respectively). The first GRU returns the full sequence, while the second returns only the final hidden state, encoding the temporal context of the entire audio clip.

c) Classification Head

Identical to the CNN model, utilizing Dense layers with Dropout and a final softmax layer.

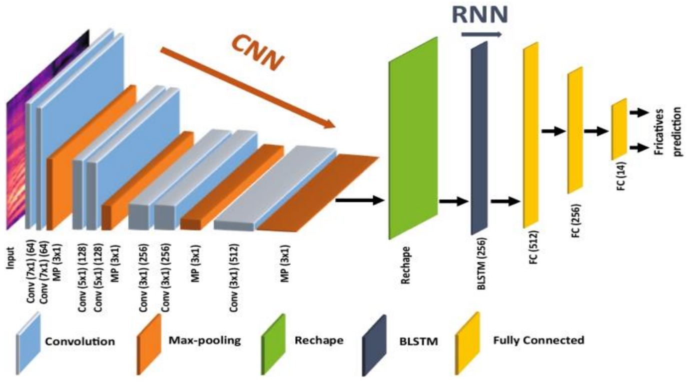


Fig. 4: Architectural diagram of the hybrid CRNN model for audio classification

D. Training and Evaluation

To ensure comprehensive and statistically reliable model assessment, two complementary evaluation strategies were implemented.

1. Stratified Train–Validation–Test Split:

The dataset was initially divided using a stratified 60–20–20 split for training, validation, and testing, respectively (**Fig 5**). This approach preserved class distribution across all subsets and enabled efficient baseline evaluation of both CNN and CRNN architectures.

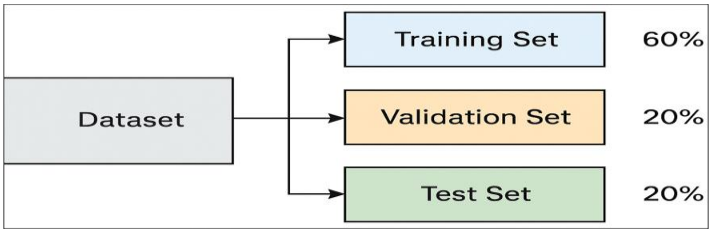
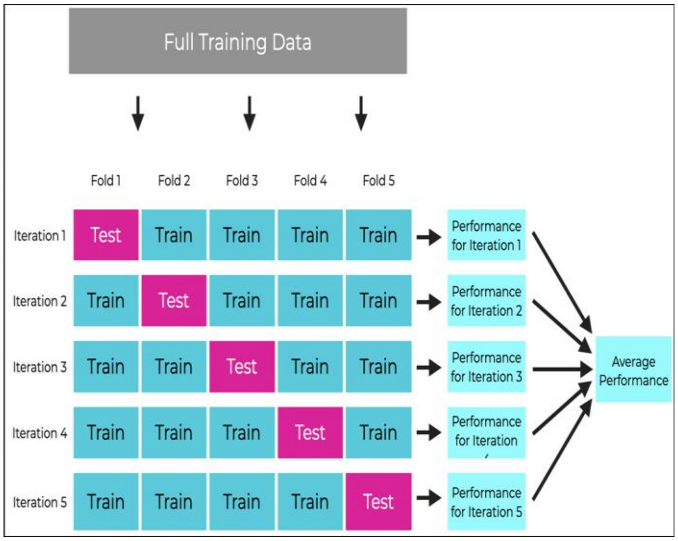


Fig 5: Distribution of the dataset into three non-overlapping subsets

2. K -Fold Cross-Validation Process:

To enhance statistical robustness and minimize potential bias resulting from a single random partition, a stratified 5-fold cross-validation procedure was employed (**Fig 6**). The dataset was divided into five equal folds while maintaining class balance. In each iteration, four folds were used for training and one for validation, and the process was repeated five times. The final performance metrics were computed as the **mean ± standard deviation** across all folds.



	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	
Iteration 1	Test	Train	Train	Train	Train	Performance for Iteration 1
Iteration 2	Train	Test	Train	Train	Train	Performance for Iteration 2
Iteration 3	Train	Train	Test	Train	Train	Performance for Iteration 3
Iteration 4	Train	Train	Train	Test	Train	Performance for Iteration
Iteration 5	Train	Train	Train	Train	Test	Performance for Iteration 5

Fig 6 : Illustration 5-Fold Cross-Validation Process

DOI:10.5281/zenodo.17553161

Both models were compiled with the **Adam** optimizer (initial learning rate=0.001) and the **categorical cross-entropy** loss function. They were trained with a batch size of 32 for a maximum of 100 epochs. To ensure robust training and prevent overfitting, key callbacks were employed:

- **EarlyStopping:**

Monitored validation accuracy with a patience of 10 epochs to halt training when performance plateaued.

- **ReduceLROnPlateau**

Reduced the learning rate by a factor of 0.2 if validation accuracy did not improve for 5 epochs.

- **ModelCheckpoint**

Saved the best-performing model based on validation accuracy.

Model performance was evaluated on the held-out test set using standard metrics: **Accuracy, Precision, Recall, and F1-Score**.

A detailed **Confusion Matrix** and **ROC-AUC** curves were also analyzed to assess class-wise performance and overall discriminative power.

III. RESULTS AND ANALYSIS

This section presents the experimental results of both the CNN and CRNN models, followed by a comprehensive comparative analysis.

A. Experimental Setup

The models were implemented using Python 3.10.0 with TensorFlow and Keras frameworks. Feature extraction was performed using Librosa library. Training was conducted on a machine with 16GB RAM and Intel HD Graphics 4600.

B. Performance Evaluation Using Stratified Split

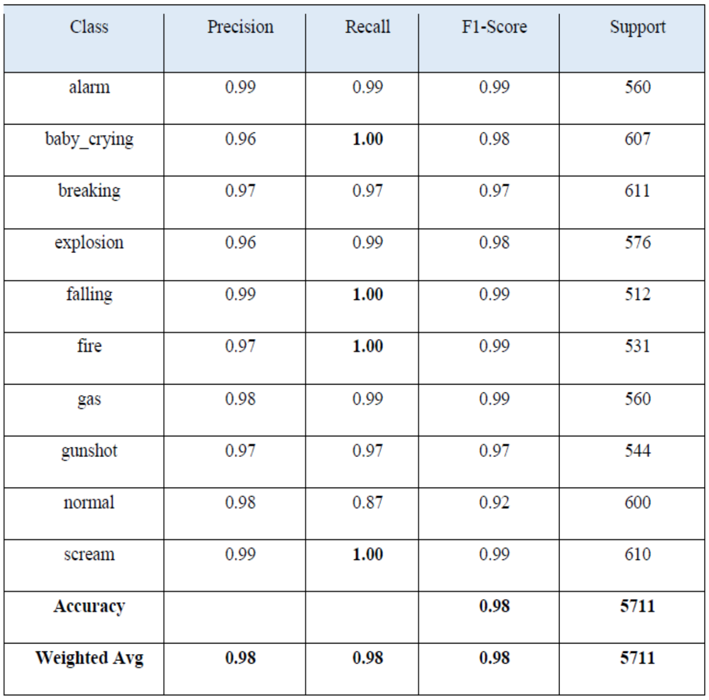
1. CNN Model Performance

The Convolutional Neural Network (CNN) model demonstrated exceptional performance in emergency sound classification, achieving state-of-the-art results across multiple evaluation metrics.

a. Quantitative Performance Metrics

As detailed in **Table 2**, the CNN model achieved an overall accuracy of 98% with a weighted average F1-score of 0.98, indicating robust classification capability across all emergency sound categories.

Table 2: Classification Report for CNN Model



Class	Precision	Recall	F1-Score	Support
alarm	0.99	0.99	0.99	560
baby_crying	0.96	**1.00**	0.98	607
breaking	0.97	0.97	0.97	611
explosion	0.96	0.99	0.98	576
falling	0.99	**1.00**	0.99	512
fire	0.97	**1.00**	0.99	531
gas	0.98	0.99	0.99	560
gunshot	0.97	0.97	0.97	544
normal	0.98	0.87	0.92	600
scream	0.99	**1.00**	0.99	610
Accuracy			**0.98**	**5711**
Weighted Avg	**0.98**	**0.98**	**0.98**	**5711**

Notably, the model achieved perfect recall (1.00) for four critical emergency classes: baby_crying, falling, fire, and scream, demonstrating its exceptional capability in detecting these crucial events without false negatives.

b. Training Behavior and Convergence Analysis

The training dynamics of the CNN model, illustrated in **Fig 7**, reveal excellent learning characteristics with rapid convergence within 15-20 epochs. The perfect alignment between training and validation accuracy curves indicates superior generalization capability, while the corresponding loss curves show stable minimization to negligible values.

c. Confusion Matrix Analysis

The confusion matrix presented in **Fig 8** provides detailed insights into class-wise performance. The strong diagonal concentration confirms effective classification across all categories. Minor misclassifications occurred primarily in the "normal" class, with 13 instances misclassified as "baby_crying" and 17 as "breaking," suggesting acoustic similarities between these categories that warrant further investigation.

DOI:10.5281/zenodo.17553161

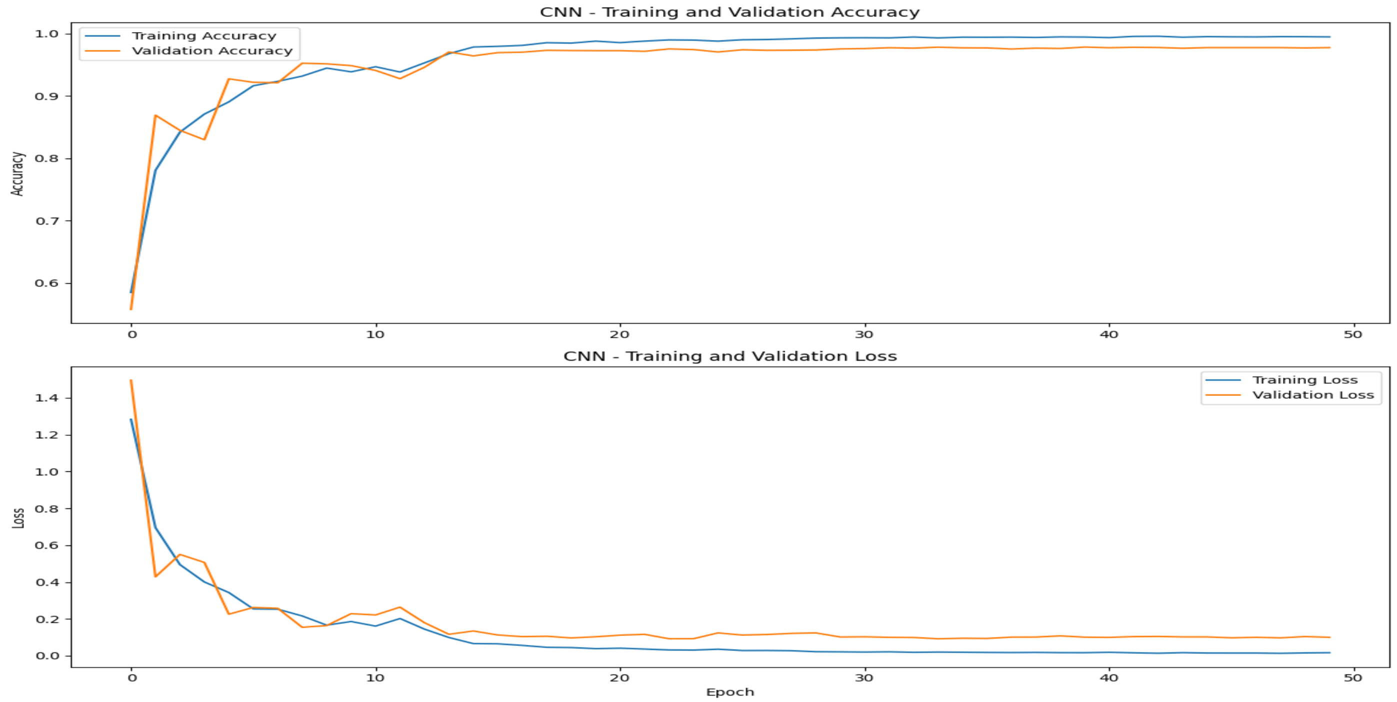
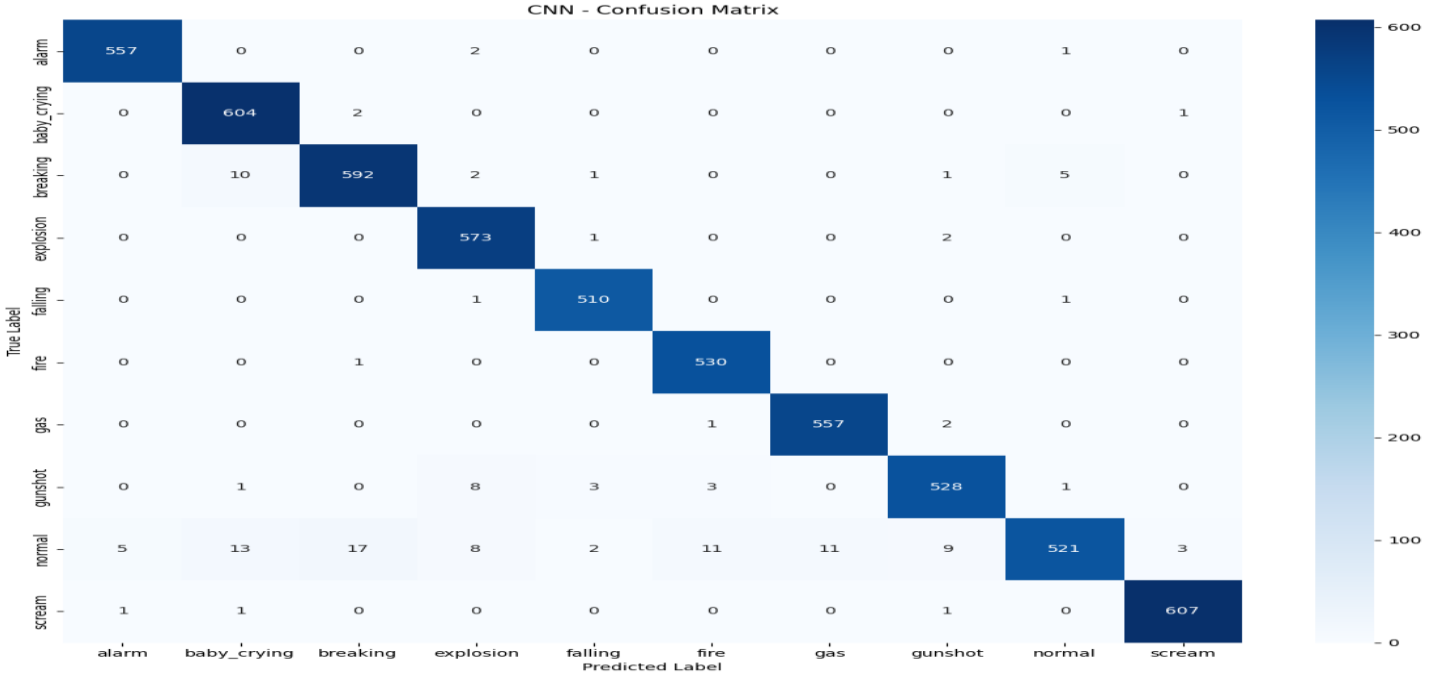


Fig. 7: CNN Training/Validation Accuracy and Loss curves



	alarm	baby_crying	breaking	explosion	falling	fire	gas	gunshot	normal	scream
alarm	557	0	0	2	0	0	0	0	1	0
baby_crying	0	604	2	0	0	0	0	0	0	1
breaking	0	10	592	2	1	0	0	1	5	0
explosion	0	0	0	573	1	0	0	2	0	0
falling	0	0	0	1	510	0	0	0	1	0
fire	0	0	1	0	0	530	0	0	0	0
gas	0	0	0	0	0	1	557	2	0	0
gunshot	0	1	0	8	3	3	0	528	1	0
normal	5	13	17	8	2	11	11	9	521	3
scream	1	1	0	0	0	0	0	1	0	607

Fig 8: CNN Confusion Matrix

DOI:10.5281/zenodo.17553161

d. Discriminative Power Assessment

The Receiver Operating Characteristic (ROC) analysis, depicted in **Fig 9**, demonstrates outstanding discriminative capability with Area Under Curve (AUC) values ranging from 0.996 to 1.000 across all classes. These near-perfect AUC scores confirm the model's superior ability to distinguish between emergency sound categories.

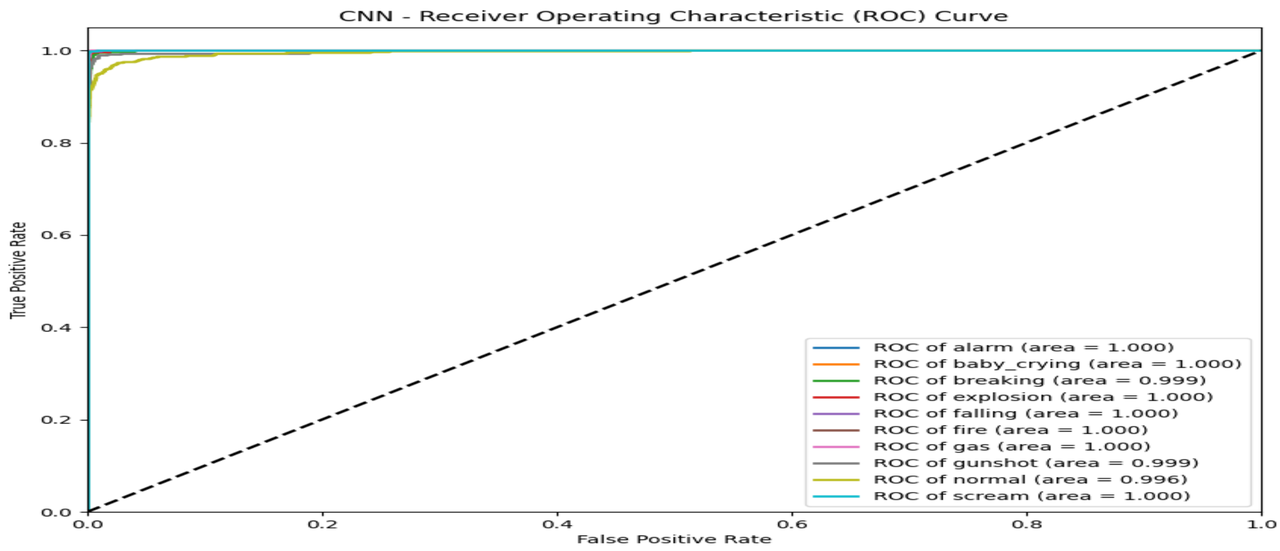


Fig 9: CNN ROC Curves with Class-wise AUC Values

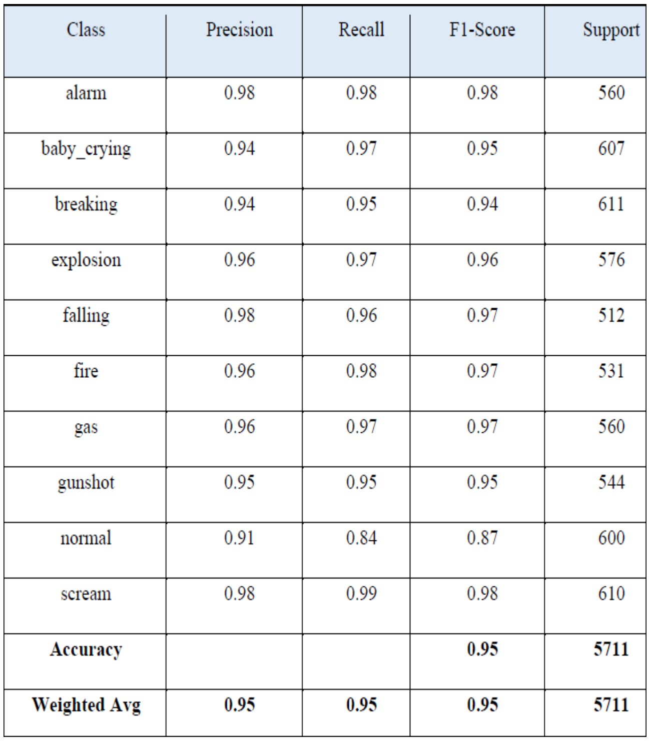
2. CRNN Model Performance

The Convolutional Recurrent Neural Network (CRNN) model showed competent performance but was consistently outperformed by the simpler CNN architecture.

a. Quantitative Performance Metrics

As summarized in **Table 3**, the CRNN model achieved an overall accuracy of 95% with a weighted average F1-score of 0.95. The CRNN model struggled most with the "normal" class, achieving the lowest recall (0.84) and F1-score (0.87) among all categories.

Table 3: Comprehensive Classification Report for CRNN Model



Class	Precision	Recall	F1-Score	Support
alarm	0.98	0.98	0.98	560
baby_crying	0.94	0.97	0.95	607
breaking	0.94	0.95	0.94	611
explosion	0.96	0.97	0.96	576
falling	0.98	0.96	0.97	512
fire	0.96	0.98	0.97	531
gas	0.96	0.97	0.97	560
gunshot	0.95	0.95	0.95	544
normal	0.91	0.84	0.87	600
scream	0.98	0.99	0.98	610
Accuracy			**0.95**	**5711**
Weighted Avg	**0.95**	**0.95**	**0.95**	**5711**

b. Training Behavior and Convergence Patterns

DOI:10.5281/zenodo.17553161

The training curves shown in **Fig 10** indicate slower convergence (30-35 epochs) compared to the CNN model. A noticeable gap between training and validation curves suggests slight overfitting tendencies, and the training process exhibited less stability throughout the learning phase.

c. Confusion Matrix Analysis

The confusion matrix in **Fig 11** reveals more frequent misclassifications compared to the CNN model, particularly for the "normal" class which was often confused with emergency categories such as "breaking," "gas," and "fire."

d. Discriminative Power Assessment

The ROC analysis in **Fig 12** shows strong discriminative performance with AUC values ranging from 0.989 to 1.000, though slightly inferior to the CNN model, particularly for the "normal" category.

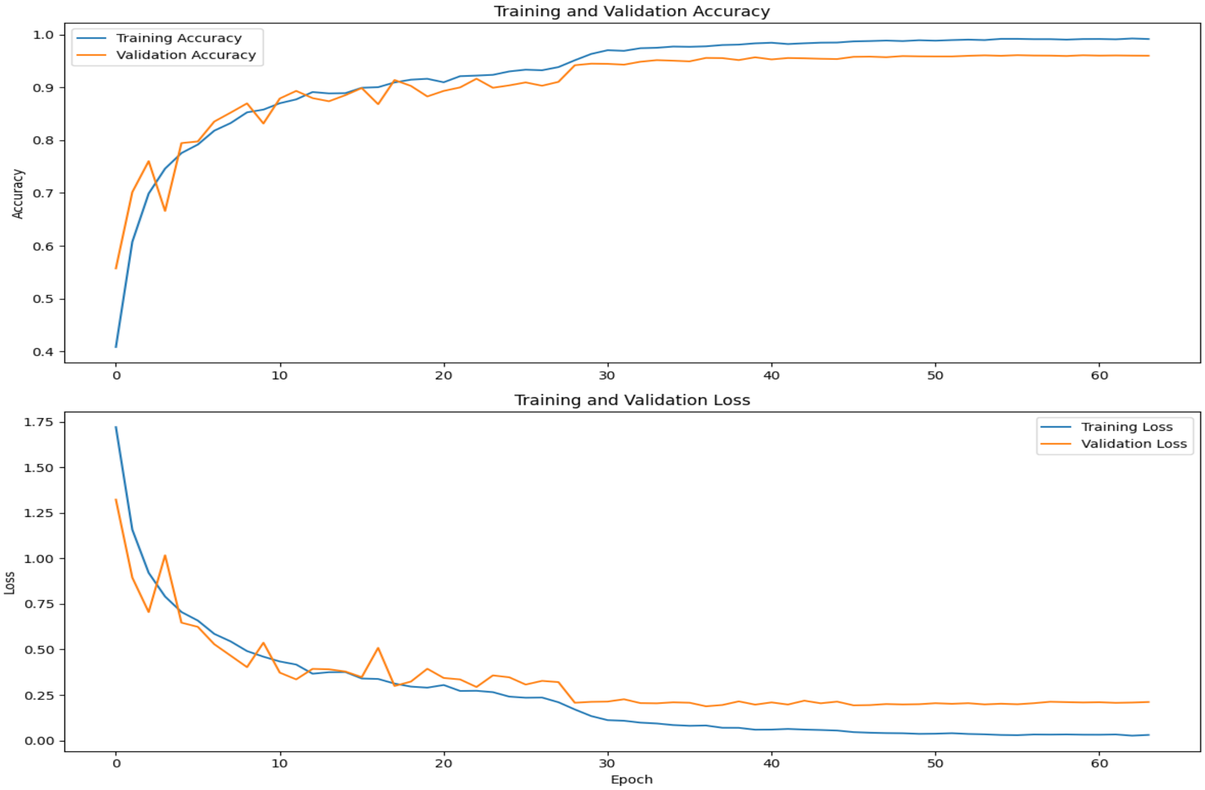
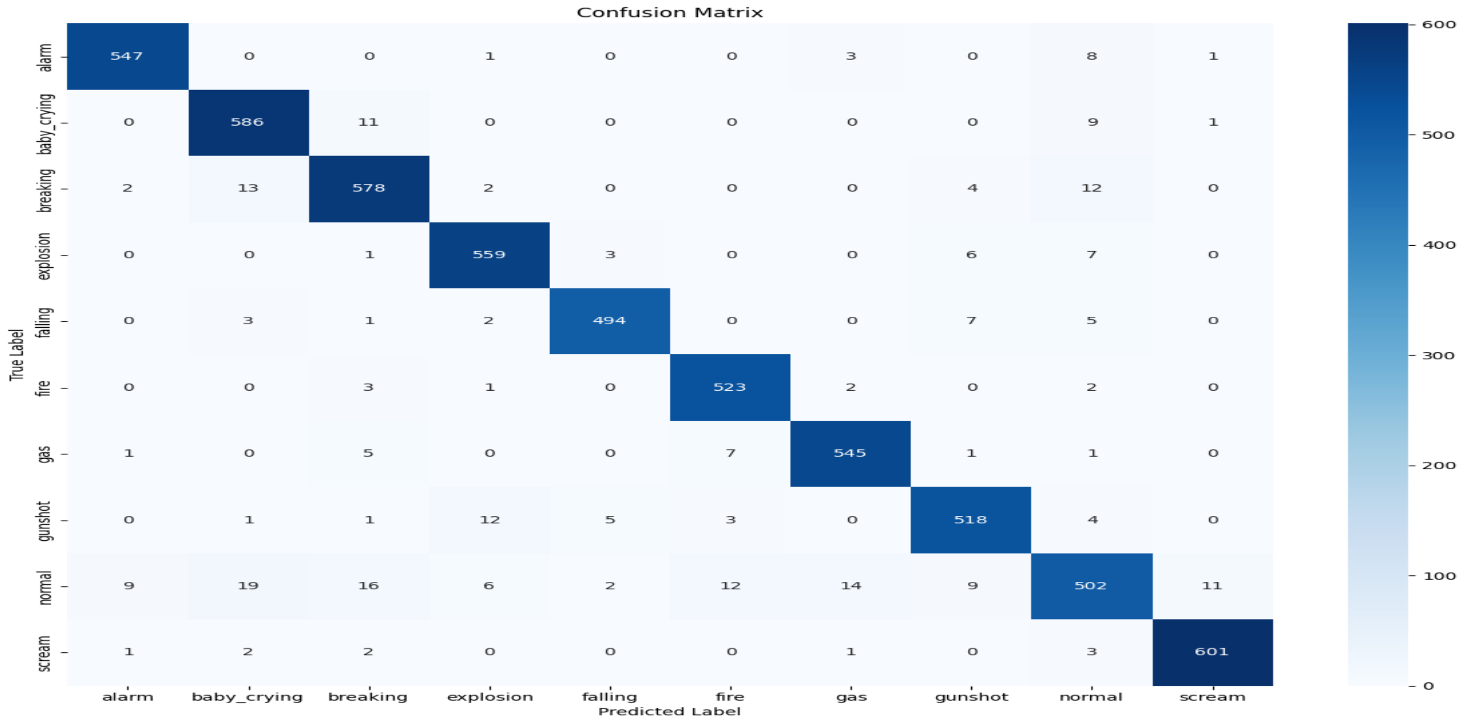


Fig 10: CRNN Training/Validation Accuracy and Loss curves

DOI:10.5281/zenodo.17553161



	alarm	baby_crying	breaking	explosion	falling	fire	gas	gunshot	normal	scream
alarm	547	0	0	1	0	0	3	0	8	1
baby_crying	0	586	11	0	0	0	0	0	9	1
breaking	2	13	578	2	0	0	0	4	12	0
explosion	0	0	1	559	3	0	0	6	7	0
falling	0	3	1	2	494	0	0	7	5	0
fire	0	0	3	1	0	523	2	0	2	0
gas	1	0	5	0	0	7	545	1	1	0
gunshot	0	1	1	12	5	3	0	518	4	0
normal	9	19	16	6	2	12	14	9	502	11
scream	1	2	2	0	0	0	1	0	3	601

Fig 11: CRNN Confusion Matrix Visualization

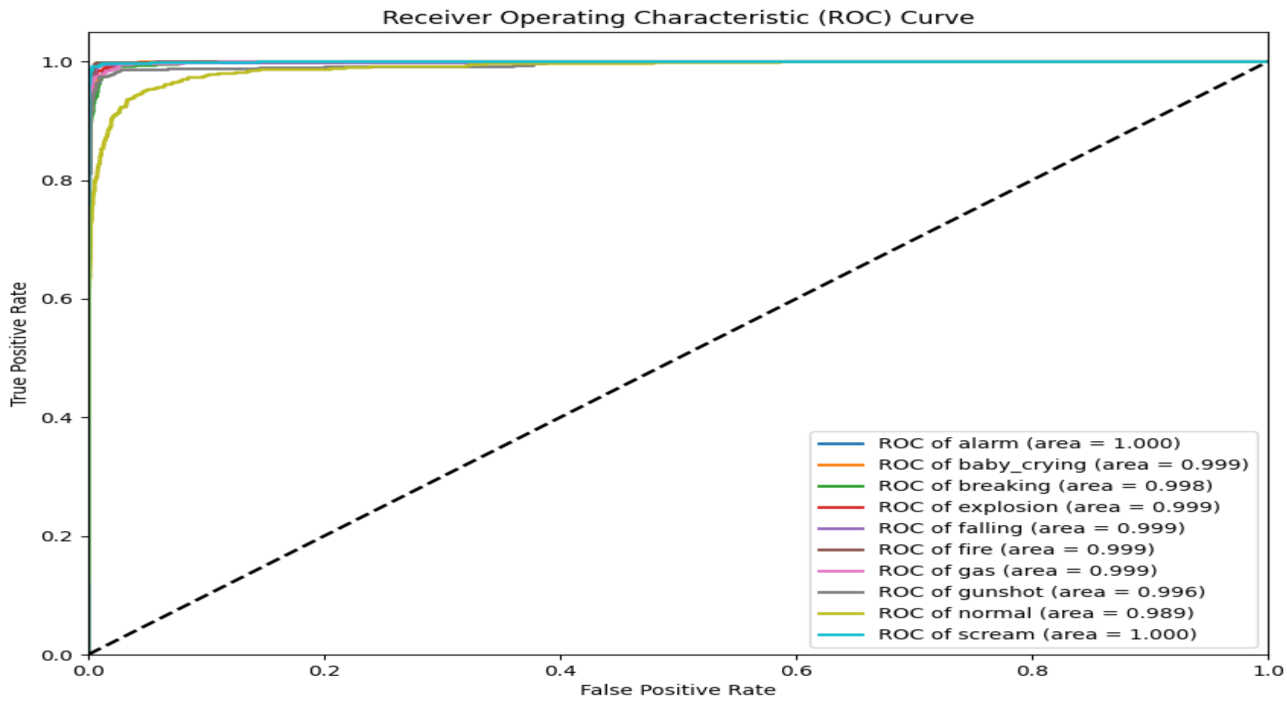


Fig 12: CRNN ROC Curves with Class-wise AUC Values

DOI:10.5281/zenodo.17553161

C. Model Performance Using 5-Fold Cross-Validation

1. CNN Model Performance

The CNN model was further evaluated using a 5-Fold Cross-Validation protocol to ensure consistency and robustness. It achieved a mean accuracy of 98%, with macro and weighted F1-scores of 0.98.

The validation accuracy and loss curves illustrated in **Fig 13 and 14** show smooth convergence with minimal fluctuation across folds.

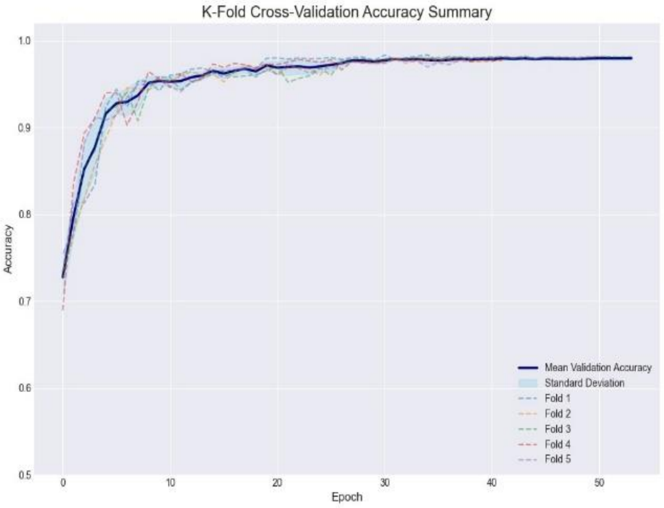


Fig13: Validation Accuracy Curves for CNN (5-Fold Cross-Validation)

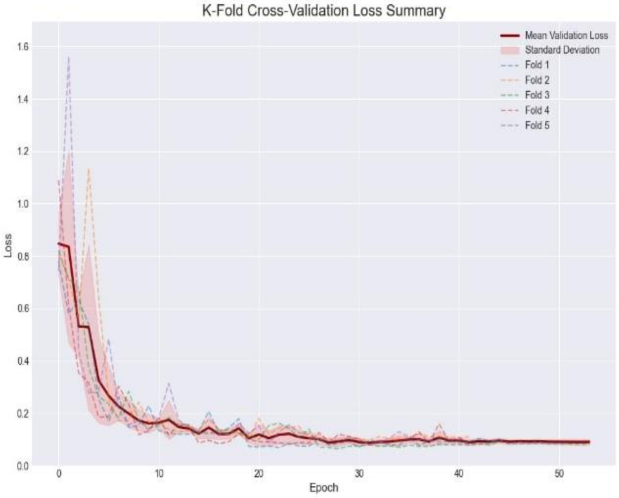


Fig 14: Validation Loss Curves for CNN (5-Fold Cross-Validation)

2. CRNN Model Performance

To provide a fair comparison, the CRNN model was evaluated under the same 5-Fold Cross-Validation protocol. **Fig 15 and 16** illustrate the validation accuracy and loss curves across folds.

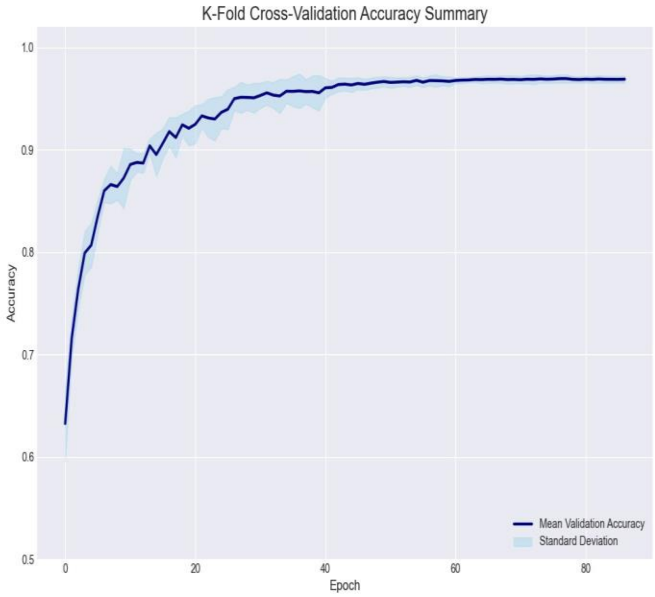


Fig 15: Validation Accuracy Curves for CRNN (5-Fold Cross-Validation)

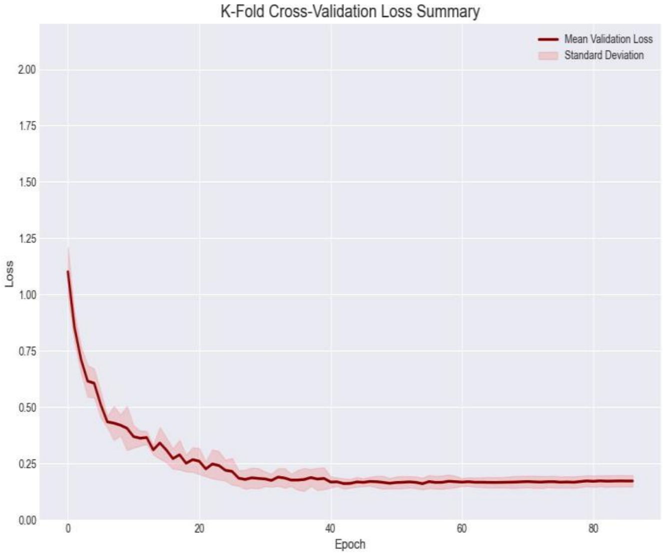


Fig 16: Validation Loss Curves for CRNN (5-Fold Cross-Validation)

Unlike the CNN, the CRNN curves show slower convergence and greater fluctuation, especially during early epochs, indicating less stable optimization and weaker generalization capability.
These variations suggest that the CRNN struggled to maintain consistent learning across different data partitions.

Despite this instability, the CRNN still achieved satisfactory results, with a mean accuracy of 89.3% and a mean loss of 0.15, as shown in **Table 4**.

However, its higher standard deviation (**±3.5%**) compared to CNN (**±0.9%**) highlights a stronger sensitivity to data variation and reduced robustness.

Table 4. Model Performance Summary (Cross-Validation)

Model	Mean Accuracy	Mean Loss	Standard Deviation
CNN	98.0%	0.07	± 0.9%
CRNN	89.3%	0.15	± 3.5%

D. Comparative Analysis and Discussion

1. Comprehensive Performance Comparison

A comparative analysis between the **stratified split** and **cross-validation** confirms the robustness of the CNN model.

The model maintained almost identical accuracy (**98.0% vs 97.8%**), demonstrating consistent learning dynamics and strong generalization capability.

Conversely, the CRNN showed a notable accuracy decline (**95.0% to 89.3%**), emphasizing its sensitivity to data segmentation.

Table 5. Performance Comparison between Evaluation Methods

Model	Stratified Split	Cross-Validation	Difference
CNN	98.0%	97.8%	−0.2%
CRNN	95.0%	89.3%	−5.7%

These results reinforce that the CNN architecture not only delivers superior accuracy but also exhibits remarkable consistency and reliability, making it more suitable for real-world emergency sound detection systems.

2. Architectural Implications and Theoretical Insights

The cross-validation findings further strengthen the theoretical interpretation of CNN’s superiority over CRNN.
Several factors explain this consistent advantage:

a) **Acoustic Nature of Target Sounds**

Most emergency sounds in our dataset (gunshots, breaking glass, screams) are short-duration events where spectral characteristics dominate temporal patterns. The CNN's spatial feature extraction proves sufficient and more efficient for these acoustic signatures.

b) **Fixed-length Segmentation Impact**

The 5-second audio standardization, while necessary for batch processing, may have reduced long-term temporal dependencies that CRNNs are designed to capture, thereby diminishing their theoretical advantage.

c) **Model Complexity and Generalization**

The CRNN's additional recurrent layers increased model complexity without commensurate performance benefits, leading to slight overfitting tendencies and reduced generalization capability compared to the more robust CNN architecture.

3. Practical Deployment Considerations

Beyond theoretical performance, the CNN architecture demonstrates several practical benefits that enhance its suitability for real-world implementation:

a) **Computational Efficiency**

Faster training convergence reduces development time and computational costs

b) **Training Stability**

More reliable learning dynamics facilitate reproducible results

c) **Resource Optimization**

Lower complexity makes the model better suited for resource-constrained smart home devices

Maintenance Simplicity

Easier hyperparameter tuning and monitoring

IV. CONCLUSION AND FUTURE WORK

A. Conclusion

Through our hands-on experimental work comparing CNN and CRNN models for home emergency sound detection, we arrived at a clear and consistent finding: the CNN architecture significantly outperforms the CRNN approach. Using both a standard stratified split and 5-fold cross-validation, the CNN achieved 98.0% accuracy under the first method and 97.8% ± 0.9% under the second, demonstrating not only high

DOI:10.5281/zenodo.17553161

performance but also remarkable stability across different data partitions.

Contrary to widespread assumptions in the literature—and indeed, our own initial expectations—we found that the more complex CRNN model, despite its theoretical strength in modeling temporal sequences, was consistently less accurate and less stable. This outcome suggests that for short-duration, spectrally distinct emergency sounds such as glass breaking, screams, or gunshots, the spatial feature extraction capabilities of CNNs are not only sufficient but actually more effective.

From a practical standpoint, our implementation experience confirmed that the CNN is also more efficient to train and easier to optimize, making it better suited for deployment in resource-constrained environments such as smart homes. Based on these results, we strongly recommend the adoption of CNN-based models for audio-based emergency detection systems where reliability and computational efficiency are critical.

B. Future Work

Building on the empirical evidence gathered in this study, we identify several meaningful directions for future research:

a) Lightweight CNN Architectures:

We intend to explore streamlined versions of the CNN, such as MobileNet or EfficientNet adaptations, to further reduce computational overhead while retaining detection accuracy—especially relevant for edge device deployment.

b) Context-Aware Temporal Modeling:

While the CRNN did not excel in our current task, we recognize that certain emergency scenarios, such as prolonged cries for help or continuous gas leaks, may still benefit from temporal modeling. Hybrid models that selectively activate temporal processing could be investigated.

c) In-Situ Real-World Testing:

An essential next step is moving beyond laboratory datasets to evaluate model performance in real home environments, where background noise, room acoustics, and overlapping sounds present additional challenges.

d) Multimodal Sensing Integration:

We plan to enrich the audio analysis pipeline by integrating non-acoustic sensors—such as motion, vibration, or smoke detectors—to reduce false alarms and improve detection confidence.

e) Cross-Environment Generalization:

Another important avenue is assessing model adaptability across varied acoustic settings (e.g., different household sizes, building materials, or cultural noise backgrounds) to ensure global applicability.

f) Few-Shot and Continual Learning:

Finally, we see value in developing incremental learning strategies that allow the system to learn new emergency sounds over time without full retraining—enabling personalized and adaptable detection systems.

References

[1] United Nations Department of Economic and Social Affairs, World Social Report 2023: Leaving No One Behind in an Ageing World. New York, NY, USA: UN, 2023.

[2] World Health Organization, "Falls," Fact Sheet, Apr.2021.[Online]. Available: https://www.who.int/news-room/fact-sheets/detail/falls

[3] M. A. Al-Qerem et al., "IoT privacy and security: Challenges and solutions," Journal of Network and Computer Applications, vol. 191, Art. no. 103209, 2021.

[4] S. R. Islam, D. Kwak, M. H. Kabir, M. Hossain, and K.-S. Kwak, "The Internet of Things for health care: A comprehensive survey," IEEE Access, vol. 3, pp. 678-708, 2015.

[5] M. M. Islam, A. Rahaman, and M. R. Islam, "Development of smart healthcare monitoring system in IoT environment," SN Computer Science, vol. 1, no. 185, 2020.

[6] J. F. Gemmeke et al., "AudioSet: A large-scale dataset of audio events," IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 31, pp. 1-14, 2023.

[7] J. Wang et al., "Advanced audio feature extraction techniques for deep learning models," Digital Signal Processing, vol. 138, Art. no. 104033, 2023.

[8] J. Kim, K. Min, M. Jung, and S. Chi, "Occupant behavior monitoring and emergency event detection in single-person households using deep learning-based sound recognition," Building and Environment, vol. 181, Art. no. 107092, 2020.

[9] R. Wolniak and W. Grebski, "The usage of smart cameras in smart home," Scientific Papers of Silesian University of Technology, Organization and Management Series, no. 188, pp. 687-700, 2023.

DOI:10.5281/zenodo.17553161

[10] N. Smailov et al., "A Novel Deep CNN-RNN Approach for Real-time Impulsive Sound Detection to Detect Dangerous Events," International Journal of Advanced Computer Science and Applications, vol. 14, no. 4, pp. 271-280, 2023.

[11] Y. Zhang, L. Wang, and D. Liu, "Privacy-preserving visual monitoring in smart homes: A review," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 1125-1140, 2022.

[12] X. Chen and H. Wang, "Wearable sensors for health monitoring: Compliance issues and solutions," *IEEE Reviews in Biomedical Engineering*, vol. 14, pp. 234-247, 2021.

[13] K. J. Piczak, "Environmental sound classification with convolutional neural networks," *IEEE Signal Processing Letters*, vol. 29, pp. 1232-1236, 2022.

[14] J. Salamon and J. P. Bello, "Deep Convolutional Neural Networks and Data Augmentation for Environmental Sound Classification," IEEE Signal Processing Letters, vol. 24, no. 3, pp. 279-283, 2017.

[15] W. Yu, S. Li, and Q. Huang, "LSTM-based audio event detection for smart home applications," Applied Acoustics, vol. 182, Art. no. 108264, 2021.

[16] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," Advances in Neural Information Processing Systems, vol. 33, pp. 12449-12460, 2020.

[17] Z. Mushtaq and S.-F. Kim, "CRNN-based multi-temporal convolution feature map for sound event detection," Applied Sciences, vol. 13, no. 4, p. 2567, 2023.

[18] Y. Liu, M. Liu, and J. Li, "A comprehensive dataset for home emergency sound detection," Scientific Data, vol. 11, no. 1, p. 345, 2024.

[19] P. Kumar and R. Singh, "Comparative analysis of CNN and CRNN architectures for audio classification," Expert Systems with Applications, vol. 234, Art. no. 121045, 2023.

[20] M. Tan and Z. Lee, "Edge computing optimization for deep learning-based audio analysis," IEEE Internet of Things Journal, vol. 11, no. 8, pp. 13456-13468, 2024.

DOI:10.5281/zenodo.17553161