Adapting Deep Learning Models for Audio Classification to Real-Time Noise Cancellation

Joseph P Thomas
University of Illinois Urbana-Champaign
Champaign, USA
jpt7@illinois.edu

Santhra Thomas
University of Illinois Urbana-Champaign
Champaign, USA
santhra2@illinois.edu

ACM Reference Format:
Joseph P Thomas and Santhra Thomas. 2024. Adapting Deep Learning Models for Audio Classification to Real-Time Noise Cancellation. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The past few years have witnessed a surge in interest surrounding the application of deep learning in audio classification, particularly within the domain of emergency sound detection. While considerable progress has been made in areas such as environmental sound classification (ESC) and emergency sound detection, there remain notable gaps in real-time noise cancellation and the overall processing of audio data in emergency scenarios. These gaps emphasis the need for innovative solutions that not only classify emergency sounds accurately but also mitigate the impact of background noise in real-time.

This study attempts to address these gaps by proposing a rare approach that combines deep learning models for both audio classification and noise reduction, with a specific emphasis on emergency audio signals. Drawing upon insights from prior research, we outline a multi-stage methodology that seamlessly integrates classification and denoising models, thereby augmenting the reliability and efficiency of emergency response systems.

Our research is motivated by the pursuit of two primary objectives: firstly, to explore the feasibility of adapting convolutional neural network (CNN) architectures for real-time noise cancellation, and secondly, to assess the potential implications of such adaptations on various domains, including industrial sensing and consumer electronics like smartwatches and headphones. By conducting a thorough analysis of existing CNN architectures and denoising techniques, we aim to devise and implement a modified CNN model tailored specifically to identify and prioritize critical auditory signals amidst noisy environments.

The successful adaptation of CNN models for real-time noise cancellation holds promise for diverse applications, ranging from enhancing user experiences in industrial settings to elevating the functionality of consumer devices. Moreover, by addressing pertinent research questions, our study contributes to the advancement of audio processing technologies, with far-reaching implications for public safety and beyond. Ultimately, our goal is to leverage cutting-edge technologies to address pressing challenges and enhance the well-being of individuals and communities alike.

2 LITERATURE REVIEW

Numerous studies have explored the use of deep learning for audio classification, but few specifically focus on emergency audio, with one of the primary papers conducted by Jivitesh Sharma et al. [1]. The literature on environmental sound classification (ESC) and emergency sound detection has seen significant advancements, with recent research focusing on developing robust systems capable of identifying various auditory cues indicative of emergencies. In this study, the authors propose a system that extends beyond the conventional detection of sirens and screams to encompass a broader spectrum of potentially hazardous environmental sounds. Through rigorous testing on three benchmark ESC datasets, both individually and in combination, the proposed system demonstrates its efficacy in binary classification between emergency and non-emergency sounds, achieving an accuracy of 99.56 percent. Leveraging multiple audio/signal processing techniques for feature extraction and a convolutional neural network (CNN) with separable convolutions to process temporal and feature domain information separately, the methodology ensures comprehensive analysis and utilization of auditory data. The research did not mention the implementation of real-time audio classification.

Devis Nugroho et al.[2] conducts a research that addresses the critical need for automated recognition and classification of alarm sounds to enhance safety for deaf individuals. By proposing a tool system that uses deep learning models, namely Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM), this study demonstrates the efficacy of these models in accurately identifying various alarm signals. Leveraging Mel-Spectrogram features extracted from an alarm audio dataset, the CNN model exhibited superior performance with an accuracy of 98.83 percent, compared to the LSTM model's accuracy of 96.66 percent. Proposing a real-time emergency sound classification tool.

There has been limited research on real-time noise cancellation. Bernd Porr et al. [3], introduces a approach utilizing a compound electrode design coupled with a deep learning algorithm to effectively eliminate noise from EEG signals, particularly targeting EMG interference. By generating an opposing signal to the noise through algorithmic means, the system achieves adaptive noise removal in real-time, without the conventional sequential process

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

of training followed by filtering. This study showcases a significant advancement in simultaneous learning and noise reduction, marking a departure from traditional methodologies. Moreover, the incorporation of a constantly adapting spatial Laplace filter further enhances the efficacy of noise cancellation. The experimental validation conducted on EEG data from 20 subjects engaging in jaw-clenching activities underscores the viability and effectiveness of the proposed approach.

3 PROBLEM STATEMENT

In today's digital age, the integration of advanced technology into wearable devices like smartwatches presents exciting opportunities for enhancing user experiences. One compelling vision is the development of a smartwatch feature capable of real-time noise cancellation, allowing users to focus on vital alerts while filtering out distracting background noise. Such an innovation would be particularly valuable during study sessions, moments of solitude, or in noisy environments where concentration is essential.

However, current noise cancellation technology primarily relies on headphones or specialized earplugs, which can be cumbersome and limiting for users seeking a seamless audio experience. The challenge is to adapt sophisticated deep learning models, originally designed for audio classification, to effectively enable real-time noise cancellation on wearable devices like smartwatches.

Recent studies by Jivitesh Sharma, Ole-Christoffer Granmo, and Morten Goodwin propose a novel approach using convolutional neural networks (CNNs) for audio classification, specifically identifying emergency situations from audio signals. Meanwhile, research by Porr et al. introduces methods for real-time noise cancellation using deep neural networks (DNNs), demonstrating simultaneous learning and noise reduction without traditional sequential training and filtering processes.

The gap in current research lies in the exploration of whether CNN models, known for their accuracy in audio classification tasks, can be adapted to Porr et al.'s real-time noise cancellation techniques. The successful integration of these approaches could revolutionize industrial sensing and consumer applications, potentially leading to innovations like noise-cancelling headphones embedded with intelligent noise reduction capabilities.

The primary objective of this study is to investigate the feasibility of adapting CNN-based audio classification models to achieve real-time noise cancellation as proposed by Porr et al. To address this, the following research questions will guide our investigation:

- How can CNN architectures be modified to incorporate real-time noise cancellation capabilities?
- What impact would the integration of CNN-based noise cancellation have on industrial sensing and consumer devices like smartwatches or headphones?
- Can this approach effectively prioritize and enhance the recognition of vital alerts (e.g., fire alarms, police sirens) in noisy environments?

Our research will involve a comprehensive analysis of existing CNN architectures for audio classification and real-time noise cancellation techniques proposed by Porr et al. We will explore the integration of multiple feature extraction methods such as MFCC to create a robust multi-channel input for the CNN classifier, as demonstrated by Sharma, Granmo, and Goodwin.

Additionally, we will investigate the feasibility of leveraging domain-wise convolutions, separable convolutions, pooling layers, and other techniques employed by Porr et al. for real-time noise reduction. The study will culminate in the design and implementation of a modified CNN architecture tailored to prioritize and enhance the detection of critical auditory signals in noisy environments.

The successful adaptation of CNN models for real-time noise cancellation holds significant implications for various applications, including improving user experiences in industrial settings and enhancing consumer devices like noise-cancelling headphones or smartwatches. Moreover, this research opens avenues for future exploration into adaptive learning rates and novel activation functions suitable for efficient signal processing and noise reduction tasks.

4 METHODOLOGY

The research methodology comprises five distinct stages aimed at effectively classifying emergency audio signals and denoising noisy recordings. The rationale behind choosing these methods lies in addressing the unique challenges posed by emergency audio processing. In the first stage, a classification model is trained, focusing on accurately identifying different types of emergency sounds. This involves using feature extraction methods such as MFCC, which enables the model to capture essential characteristics of emergency signals. Training the model on labeled data ensures its ability to learn intricate patterns within audio data, crucial for accurate classification. The second stage involves deploying this trained model for real-time inference on new audio samples, assessing its performance metrics to validate its effectiveness in practical scenarios. Following this, the third stage focuses on training a separate model specifically for denoising noisy audio recordings. Various denoising techniques are explored, including deep learning-based methods, to mitigate the adverse effects of background noise on classification accuracy. By training the denoising model using a combination of clean and noisy audio data, the research aims to identify the most effective strategy for enhancing signal clarity. In the fourth stage, the trained denoising model is applied to noisy audio inputs to improve signal quality before classification. Finally, in the fifth stage, the outputs of the classification and denoising models are integrated, ensuring a comprehensive approach to address both classification and denoising aspects of emergency audio processing. This multi-stage methodology aims to improve the reliability and efficiency of emergency response systems, ultimately contributing to enhanced public safety outcomes.

5 EXPERIMENTAL SETUP

5.1 Dataset

The dataset utilized is from Jivitesh Sharma et al. [1] where this research comprises 3600 audio samples capturing various sounds emitted by real alarm equipment. Ninety percent of this dataset is manually created, while the remaining 10 is sourced from diverse internet repositories. The creation process involves recording alarm sounds directly from the devices using a microphone connected to a Raspberry Pi 4. The dataset is categorized into five distinct classes: fire alarm (FA), gas leak alarm (GA), tsunami alarm (TA), danger alarm (DA), and background noise (BN). Each class consists of a specific number of samples, with exactly 700 samples for FA, GA, TA, and DA, and 800 samples for BN to account for its varied nature. All audio recordings are stored in WAV format to preserve optimal audio quality throughout the dataset.

The distribution of samples across classes is summarized in Table I, providing insights into the balanced nature of the dataset. Researchers can access and download the collected dataset through the provided link (https://www.kaggle.com/datasets/devisdesnug/alarmdataset/download).

A modified version of the dataset for classes FA, GA, TA, DA has been created for the purpose of training an audio denoising model which adds another 700 samples to the dataset. This modified dataset includes additional noise added to the audio samples, simulating common background noises such as crowd noise, machinery, traffic, music, and appliance sounds. The noisy audio dataset comprises 10 distinct audio samples, each distinct from the common background noise prevalent in emergency scenarios. This augmented dataset enables the training and evaluation of the denoising model under realistic conditions, enhancing its efficacy in real-world applications.

5.2 Preprocessing

In preprocessing stage, the audio dataset for audio classification undergoes crucial steps to ensure suitability for classification tasks. The dataset is loaded with the first 3000 samples considered for efficiency. A DataFrame organizes the data for systematic analysis. Exploratory data analysis checks class distribution for balance. Audio samples are verified by playing a subset. Mel-Frequency Cepstral Coefficients (MFCCs) are extracted from each audio sample, capturing crucial spectral features. Audio samples are either padded with silence or truncated to precisely one second to standardize duration. Labels are one-hot encoded for model compatibility. Finally, the dataset is split into training, validation, and test sets to ensure unbiased evaluation.

In the preprocessing stage of the audio datasets it undergo crucial steps to ensure suitability for training the denoising autoencoder model. Each audio file is loaded and it is padded or truncated to a target length to ensure uniformity across samples. A target length of 16000 samples is chosen, corresponding to 1 second of audio at a sampling rate of 16 kHz. This step is crucial for ensuring consistency in audio duration, which is essential for subsequent processing and model training. The clean and noisy audio datasets are loaded separately from their respective directories, ensuring that the audio samples are correctly associated with their corresponding labels. Finally, the audio data is reshaped to include a channel dimension, which is necessary for compatibility with the denoising autoencoder model architecture. These preprocessing steps ensure that the audio data is appropriately formatted and standardized, laying the foundation for effective model training and denoising performance.

5.3 Audio Classification

5.3.1 *Extracting Audio Features.* .

When dealing with raw audio data, such as waveforms, their direct utilization as input for convolutional neural network (CNN) models proves inadequate due to the limited information they provide, primarily focusing on sound amplitude and loudness. To enrich this input data for more comprehensive analysis, we employ a transformation process that shifts the representation from a time-based domain to a time-frequency domain.

This transformation is achieved through a Fast Fourier Transform (FFT), which facilitates the conversion of the entire audio signal into the frequency domain. However, such a transformation results in the loss of valuable temporal information regarding the signal's frequency distribution. To address this, we adopt the Short-Time Fourier Transform (STFT) technique, where the signal is divided into smaller segments or frames.

$$X_{STFT}[k,n] = \sum_{m=0}^{N-1} x[m+nH]w[m]W_N^{km}$$

Each frame undergoes the Fourier Transform independently, preserving temporal information within each segment. The STFT computation involves a sliding window concept, where the signal is divided into overlapping windows, enabling a more granular analysis. Mathematically, this process involves summation over each window, incorporating parameters such as window length, time frame index, frequency index, and overlap length.

Following the segmentation of the signal into frames, the subsequent step involves applying a window function to each frame to mitigate spectral leakage and counteract the FFT's assumption of infinite data. The Hamming window function, defined by a specific equation, is commonly used for this purpose.

$$w[n] = 0.54 - 0.46cos(\frac{2\pi n}{N-1})$$

By applying this function, we ensure a smoother transition at the edges of each frame, facilitating more accurate frequency analysis.

Subsequently, the signal undergoes computation with the FFT, resulting in a frequency domain representation.

$$X[k] = \sum_{n=0}^{N-1} x[n]e^{-j2\pi kn/N}$$

This process entails calculating the discrete Fourier Transform for each frame, yielding frequency spectra. Notably, to compute the power spectrum (periodogram), the absolute value squared of the FFT output is divided by the window length.

$$P = (\frac{|FFT(X_i)|^2}{N})$$

This step provides insights into the power distribution across different frequency components within each frame.

In the final stage, Mel filtering is employed to extract meaningful frequency bands from the power spectrum. This involves applying triangular filters, based on the Mel scale, to mimic the human auditory system's perception of sound frequency. The Mel scale, characterized by a logarithmic relationship between frequency and perceived pitch, better aligns with human auditory capabilities. By converting the power spectrum into Mel scale representations

$$m = 2595log_{10}(I + \frac{f}{700})$$

we obtain filter banks that partition the frequency domain into distinct bands. These bands capture essential frequency features relevant to human perception, facilitating a more nuanced analysis of the audio signal. Ultimately, this comprehensive process yields a 2D image representation of the audio signal, known as a spectrogram, wherein time is represented along the x-axis, frequency along the y-axis, and amplitude power as color intensity. This spectrogram provides a rich and detailed depiction of the audio signal, enabling further analysis and interpretation for various applications.

5.3.2 Neural Network Architecture. .

For the task of sound alarm classification, we use deep learning architecture: Long Short-Term Memory (LSTM) networks. Following the extraction of spectral features from the audio signals, we are left with time series data, which serves as the input for the proposed models.

LSTM Model:
LSTM, a member of the Recurrent Neural Network (RNN) family, comprises memory cells interconnected to form a neural network. Each cell integrates three gates—input, output, and forget gates—that facilitate read, write, and reset functions, respectively. Input cells receive inputs multiplied by the input gates, while output cells multiply their output by the output gate. Previous cell values are adjusted by the forget gate. The LSTM model utilized in this study incorporates two LSTM layers followed by three fully connected layers. These LSTM layers consist of 128 and 64 units, respectively, while the fully connected layers contain 64, 32, and 16 neurons. The final layer comprises five units, each corresponding to a sound class. Dropout with a rate of 25 percent is applied to mitigate overfitting.

5.4 Audio Denoising

5.4.1 Extracting Audio Features. .

The initial step in our research involves extracting essential features from audio data to facilitate subsequent analysis and modeling tasks. Raw audio data, represented as waveforms, lack the necessary structure and richness for direct utilization in machine learning models. Therefore, we use various signal processing techniques to transform the raw audio into feature-rich representations suitable for deep learning models.

The first stage of audio feature extraction involves loading audio files from specified directories using the loadaudiofiles function. This function employs the popular librosa library to ensure uniform audio duration across all files. By padding or truncating audio signals to a predefined target length, typically set at 16,000 samples, we achieve consistency in input dimensions, facilitating subsequent processing steps.

Once the audio files are loaded and preprocessed, we perform spectral analysis to extract frequency-domain features from the audio signals. Spectral analysis enables us to capture important characteristics such as pitch, timbre, and spectral envelope. We employ the Short-Time Fourier Transform (STFT) technique to transform the audio signals from the time domain to the time-frequency domain. This technique partitions the audio signals into short segments or frames, allowing us to analyze their frequency content over time.

In addition to traditional spectral analysis, we compute Mel-Frequency Cepstral Coefficients (MFCCs), a widely used feature representation in speech and audio processing tasks. MFCCs mimic the human auditory system's response to sound by capturing the distribution of energy across different frequency bands. This process involves applying a series of transformations, including Mel scaling, discrete cosine transformation (DCT), and logarithmic compression, to the power spectrum obtained from the STFT.

The extracted audio features, including STFT spectrograms and MFCCs, serve as input representations for subsequent modeling tasks. These features capture both temporal and spectral information, enabling deep learning models to learn discriminative patterns for various audio classification tasks. To ensure compatibility with convolutional neural network (CNN) and recurrent neural network (RNN) architectures, we reshape the feature tensors to include channel dimensions, facilitating seamless integration into the model pipelines.

With the extracted audio features prepared, we proceed to train and evaluate deep learning models for specific audio classification tasks. Leveraging frameworks such as TensorFlow and Keras, we design and train convolutional neural networks (CNNs) and recurrent neural networks (RNNs) to classify audio signals into predefined categories. During training, we monitor key performance metrics such as loss and accuracy to assess model convergence and generalization capabilities.

By leveraging techniques such as spectral analysis and MFCC computation, we obtain rich representations of audio signals suitable for deep learning-based classification tasks. These extracted features serve as the foundation for building robust and accurate audio classification systems capable of handling diverse real-world scenarios.

5.4.2 Neural Network Architecture. .

The denoising autoencoder model used in this research endeavors to tackle the challenge of enhancing the clarity and fidelity of audio signals contaminated by various forms of noise. Comprising an encoder-decoder architecture, the model is meticulously designed to learn and reconstruct meaningful features from corrupted audio inputs. At the outset, the encoder component initiates the processing pipeline by accepting audio samples, which are shaped according to the length of the input audio. Through the utilization of a convolutional layer with 64 filters and ReLU activation, the encoder adeptly extracts hierarchical features from the input audio, capturing essential patterns amidst the noise. Subsequent max-pooling operations are then employed to downsample the features, ensuring the retention of crucial information while reducing computational complexity.

The decoder component takes over the task of reconstructing the denoised audio signals. Beginning with the flattening of the encoded features, the decoder prepares the feature representation for reconstruction. Leveraging a sequence of convolutional and upsampling layers, the decoder meticulously reconstructs the denoised audio, aiming to faithfully restore the original signal while mitigating the effects of noise. The culmination of the decoder's efforts is marked by a convolutional layer equipped with a single filter and sigmoid activation, which generates the final denoised audio output with enhanced clarity and reduced noise artifacts.

The denoising autoencoder model learns to minimize the mean squared error between the reconstructed denoised audio and the corresponding clean input audio. By iteratively adjusting its parameters through optimization techniques such as the Adam optimizer, the model gradually improves its ability to effectively denoise audio signals. The training process involves presenting pairs of clean and noisy audio samples, allowing the model to learn the intricate relationships between input features and target outputs.

The model's performance is rigorously evaluated to gauge its efficacy in noise reduction. The computed loss metric, reflecting the mean squared error between the reconstructed and clean audio signals, serves as a critical indicator of the model's accuracy in denoising. Through meticulous analysis of the loss values and qualitative assessment of denoised audio outputs, the effectiveness of the denoising autoencoder model is comprehensively assessed.

The denoising autoencoder model stands as a powerful tool for enhancing the quality of audio signals plagued by noise. Through its sophisticated architecture, iterative training regime, and rigorous evaluation, the model demonstrates significant promise in real-world applications requiring noise-robust audio processing.

5.5 Performance Metrics

Evaluating the performance of a model represent a important aspect of this research, as the results obtained from these assessments serve as deciding factors in selecting the optimal model. Among many techniques used to measure the model performance, particularly in classification systems, lies the confusion matrix. This evaluation method facilitates the computation of a classification process's accuracy and correctness. The confusion matrix comprises four fundamental terms: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). TP denotes positive data correctly predicted, while TN represents negative data correctly predicted. Conversely, FP signifies negative data misclassified as positive, and FN indicates positive data erroneously classified as negative. Leveraging this matrix, key performance metrics such as accuracy, precision, and recall can be derived. These metrics serve as indispensable yardsticks for assessing the effectiveness of the classifier or algorithm utilized in making predictions.

Accuracy serves as a fundamental testing method, quantifying the degree of proximity between predicted and actual values. By discerning the proportion of accurately classified data, accuracy provides insights into the predictive prowess of the model. The accuracy metric is formulated by summing TP and TN and dividing it by the total number of predictions, encompassing TP, TN, FP, and FN.

Precision, on the other hand, offers a nuanced assessment by comparing the relevant information obtained by the system against the total relevant information retrieved, inclusive of both relevant and irrelevant data. Precision delineates the precision of the positive predictions made by the model, reflecting its ability to avoid false positives. The precision metric is calculated by dividing TP by the sum of TP and FP.

Recall, assesses the model's performance but from a slightly different perspective. It compare the relevant information acquired by the system against the total relevant information present in the dataset, whether retrieved by the system or not. This metric clarifies the model's capacity to capture all relevant instances within the dataset, delineating its ability to minimize false negatives. Recall is derived by dividing TP by the sum of TP and FN.

In summation, accuracy, precision, and recall serve as indispensable tools in quantifying the performance of classification models, providing important insights into their predictive capabilities and highlighting areas for potential improvement.

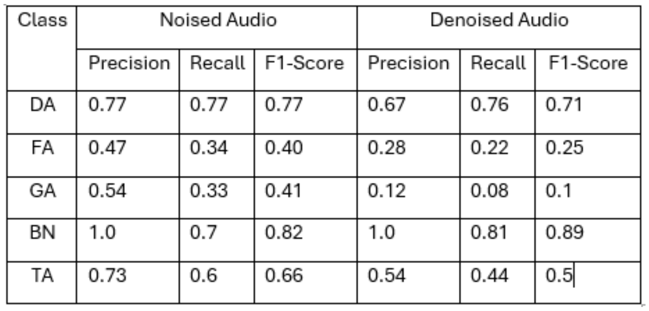
6 RESULTS AND DISCUSSION

In this experiment, we trained the model on a computer with the following specifications: For the CPU, we used Intel Core i7-3540M generation. The implementation was done in Python using the TensorFlow framework, and for converting the model to TensorFlow Lite for deployment.

The classification model architecture is constructed around a recurrent neural network (RNN), specifically utilizing Long Short-Term Memory (LSTM) units. The model architecture consists of an LSTM layer with 256 units, followed by dropout regularization to mitigate overfitting. This is succeeded by two densely connected layers with 128 and 64 units, respectively, employing ReLU activation functions and additional dropout layers. The final dense layer, activated by a softmax function, facilitates multi-class classification with 5 output units corresponding to the alarm types.

Training the model involves optimizing the categorical cross-entropy loss function using the Adam optimizer over 50 epochs with a batch size of 32. Throughout training, the model's performance is monitored on both the training and validation sets. Post-training evaluation encompasses metrics such as training, validation, and test accuracy and loss. These metrics offer insights into the model's generalization capability and performance on unseen data.

The dataset was divided into two subsets for experimentation: the training set and the testing set. The training set contained 90 percent of the entire dataset, while the testing set contained the remaining 10 percent. The training set is further divided into training and validation sets with a validation set size of 20 percent of the original dataset. We trained using cross-entropy as our loss function and the Adam Optimizer.



Class	Noised Audio			Denoised Audio		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score
DA	0.77	0.77	0.77	0.67	0.76	0.71
FA	0.47	0.34	0.40	0.28	0.22	0.25
GA	0.54	0.33	0.41	0.12	0.08	0.1
BN	1.0	0.7	0.82	1.0	0.81	0.89
TA	0.73	0.6	0.66	0.54	0.44	0.5

Figure 1: Table I

The denoising model uses a convolutional neural network CNN based architecture. It comprises a 1D convolutional layer with 64 filters, each scanning a small window of the input audio samples. These filters are convolved with the input signal, detecting patterns and features that are essential for denoising.

Activation functions, specifically Rectified Linear Unit (ReLU), are applied to introduce non-linearity, enhancing the model's capacity to capture complex relationships within the data. Additionally, padding is employed to ensure that the spatial dimensions of the input are preserved throughout the convolutional operations.

The decoder mirrors the structure of the encoder, featuring another 1D convolutional layer with 64 filters. This layer applies convolutional operations to upsample the encoded features, gradually restoring the spatial dimensions lost during encoding. Again, ReLU activation functions are utilized to introduce non-linearity and facilitate the learning of complex mappings between encoded and decoded representations.

After the training of both models, the classification model is used to predict audio with added noise, achieved by merging background noise with the data used during denoising training.The precision and recall metrics are presented in Table I for evaluation.

The denoising model is then used to remove the added noise from the data utilized during denoising training, and the precision and recall metrics are displayed in Table I.

The analysis of the results indicates that the classification performance on audio with added noise surpasses that of denoised audio. However in the classification of background noise(BN), the denoised audio demonstrates superior performance than audio with noise. This discrepancy may stem from the denoising process inadvertently eliminating crucial audio features or patterns, leading to challenges in classification. Additionally, the denoising process may inadvertently decrease the audio's volume, resulting in subtle features and patterns becoming indiscernible. Furthermore, the limited training duration of 15 epochs may contribute to imperfect classification, as the model might not have fully learned the intricate features of the audio data.

7 CONCLUSIONS

In conclusion, the investigation into adapting Convolutional Neural Network (CNN) architectures for real-time noise cancellation in wearable devices has yielded valuable insights into enhancing user experiences, particularly in scenarios where auditory alerts are crucial, such as emergency situations. This study employed a rigorous approach, combining experimentation and analysis to explore the feasibility and efficacy of CNN models for both audio classification and denoising tasks.

The classification model, based on a recurrent neural network (RNN) with Long Short-Term Memory (LSTM) units, demonstrated commendable performance in identifying various types of emergency sounds. Figure 2 depicts the training accuracy of the model across different alarm types over a series of epochs. The x-axis represents the number of epochs, which refers to the number of times the model iterates through the training data. The y-axis represents the training accuracy, which indicates the model's performance in correctly classifying alarms on the training set. As the number of epochs increases, the training accuracy trends upward for all alarm types, suggesting that the model is effectively learning to distinguish between different alarm categories.

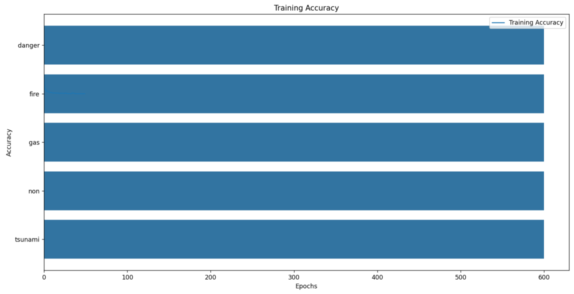


Figure 2: Training Classification Accuracy

Figure 3 depicts the training and validation loss of the model as the number of training epochs increases. The x-axis represents the number of epochs, and the y-axis represents the loss value. The training loss curve (solid line) signifies the model's performance on the training data, and ideally, it should decrease as the model learns. The validation loss curve (dashed line) reflects the model's performance on unseen validation data. A significant gap between the two curves would indicate overfitting, where the model performs well on training data but struggles to generalize to new data. In this figure, we observe a decrease in both training and validation loss as epochs progress, suggesting the model is effectively learning without overfitting. Despite its success, challenges arose when testing the model with audio containing added noise. The decrease in accuracy compared to denoised audio highlights the complexity of preserving essential audio features during the denoising process. Additionally, the impact of the limited training duration of 15 epochs on classification accuracy underscores the importance of optimizing training procedures for such models.

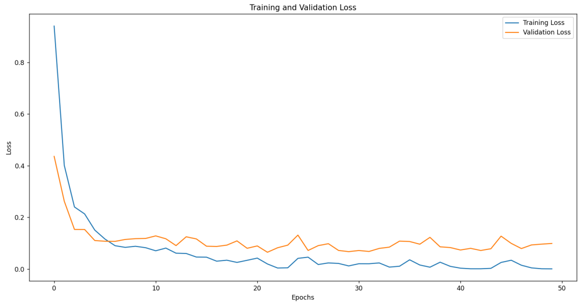


Figure 3: Training Classification Loss

Conversely, the denoising model, built on a CNN architecture with convolutional layers and ReLU activation functions, showcased effective noise reduction capabilities, particularly in eliminating background noise (BN). However, the denoising process's inadvertent effects on classification accuracy, such as altering subtle features or reducing volume, underscore the need for careful consideration of trade-offs between noise reduction and feature preservation.

The integration of classification and denoising models presents a promising avenue for enhancing emergency audio processing systems. By systematically training classification models to accurately identify different types of emergency sounds and deploying denoising models to mitigate background noise, this study offers a comprehensive approach to improving the reliability and efficiency of emergency response systems. Furthermore, the analysis of classification and denoising model performance highlights the intricate interplay between these tasks and the potential challenges encountered.

The significance of these findings extends beyond the realm of emergency response. The successful adaptation of CNN architectures for audio classification, combined with real-time noise cancellation techniques, holds implications for various applications, including industrial sensing and consumer electronics. By integrating CNN-based noise cancellation into wearable devices, users can benefit from enhanced auditory experiences, focusing on vital alerts while filtering out distracting background noise.

The methodology employed in this study, structured into distinct stages, emphasizes the importance of systematic approaches in addressing the complexities of emergency audio processing. Future research endeavors should focus on optimizing denoising processes to ensure minimal loss of important audio characteristics. Additionally, exploring adaptive learning techniques and novel architectures could further enhance the efficiency of signal processing and noise reduction tasks.

Figure 4 illustrates the training and validation loss of the denoising autoencoder model as the number of training epochs increases. The x-axis represents the number of epochs, and the y-axis represents the loss value. The training loss curve (solid line) signifies the model's performance on the training data, and ideally, it should decrease as the model learns. The validation loss curve (dashed line) reflects the model's performance on unseen validation data. A significant gap between the two curves would indicate overfitting, where the model performs well on training data but struggles to generalize to new data. In this figure, we observe a decrease in both training and validation loss as epochs progress, suggesting the model is effectively learning without overfitting. In summary, this research contributes to advancing the field of audio signal processing and holds promise for improving public safety outcomes in various real-world scenarios. By bridging the gap between audio classification and real-time noise cancellation, this study opens avenues for innovation and future exploration, ultimately aiming to revolutionize user experiences and enhance public safety outcomes.

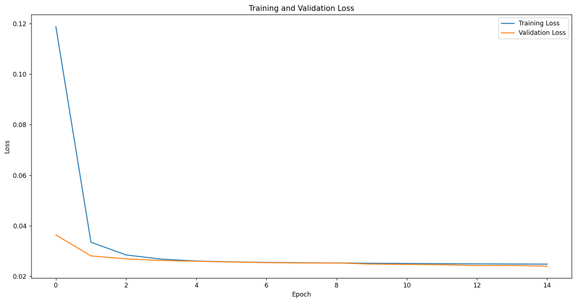


Figure 4: Training Denoise Loss

8 FUTURE WORK

8.1 Future Research Directions

In light of the findings and insights gained from this study, several promising avenues for future research emerge. These directions aim to further enhance the efficacy and applicability of the proposed methodologies, thereby advancing the field of audio signal processing and its practical applications.

Exploration of Advanced Denoising Techniques: While the convolutional neural network (CNN)-based denoising architecture employed in this study has shown promising results, future research could delve into exploring more sophisticated denoising techniques. Techniques such as recurrent neural networks (RNNs) or attention mechanisms could be investigated to further enhance the clarity of audio signals in noisy environments. Additionally, exploring the combination of multiple denoising methods or ensembling models could potentially yield superior denoising performance. Optimization for Real-Time Processing: Given the focus on real-time noise cancellation for wearable devices like smartwatches, future work could involve optimizing the model architecture for efficient inference on resource-constrained devices. This may entail exploring lightweight neural network architectures, quantization techniques, or model compression methods to reduce computational complexity and memory footprint while maintaining performance. Incorporation of Adaptive Learning Strategies: To address challenges related to imperfect classification and incomplete learning of intricate audio features, future research could investigate adaptive learning strategies. Techniques such as curriculum learning or online learning could be explored to improve the model's ability to learn complex audio patterns over time. Enhancement of Feature Extraction Methods: While Mel-Frequency Cepstral Coefficients (MFCCs) have been employed for feature extraction in this study, future research could explore alternative feature representations or combinations thereof to capture more nuanced characteristics of audio signals. Investigating the integration of additional feature extraction methods, such as Gammatone Frequency Cepstral Coefficients (GFCC) or Constant Q Transform (CQT), may provide complementary information and improve classification and denoising performance. Addressing Generalization Challenges: The observed discrepancy in performance between audio with added noise and denoised audio suggests challenges in generalization. Future research could focus on addressing these generalization issues by augmenting the dataset with diverse noise types and levels, exploring domain adaptation techniques, or employing regularization methods to enhance generalization performance.

8.2 Applications and Extensions

Beyond the realm of research, the methodologies and findings presented in this study have significant implications for various real-world applications. Here, we discuss potential avenues for applying and extending the research outcomes:

Industrial Sensing Applications: The accurate classification of emergency audio signals and effective noise reduction techniques developed in this study hold great potential for industrial sensing applications. By adapting the models for industrial environments and integrating them into monitoring systems, it could enhance safety protocols and facilitate timely responses to emergency situations in industrial settings. Consumer Electronics: The development of real-time noise cancellation capabilities demonstrated in this study is highly relevant for consumer electronics such as noise-canceling headphones or smartwatches. By leveraging the models, manufacturers could implement intelligent noise reduction features in their products, offering users a more immersive audio experience and improved usability in noisy environments. Healthcare Monitoring: The accurate detection of specific audio signals, such as medical alarms or patient distress calls, is crucial in healthcare monitoring systems. By adapting the classification models developed in this study for healthcare settings and integrating them into monitoring devices, it could aid in early detection of critical events and improve patient care outcomes. Smart Home Automation: The denoising techniques developed in this study could be valuable for smart home automation systems, particularly in scenarios where background noise interferes with audio-based commands or alerts. By integrating the denoising models into smart home devices like voice assistants or security systems, it could enhance the reliability and responsiveness of these systems in noisy household environments. Environmental Monitoring: The methodologies developed for audio classification and denoising have implications for environmental monitoring applications, such as wildlife conservation or urban noise pollution analysis. By adapting the models for environmental audio processing tasks, it could enable more accurate detection of environmental sounds and facilitate data-driven decision-making in environmental management and conservation efforts. In conclusion, the research presented in this paper not only contributes to advancements in audio signal processing but also opens up avenues for practical applications across various domains. By further exploring the suggested research directions and extending the applications of the developed methodologies, we can continue to make meaningful contributions to both academia and industry.

9 TEAM CONTRIBUTIONS

9.1 Santhra Thomas: Project Leader - 50%

- Responsible for overall project management. Made sure work from teammates supported research objectives and problem statement while making sure milestones were also accomplished on time.
- Acted as the primary point of contact for advisors or stakeholders.
- Oversaw the drafting, review, and submission of project documents.
- Participated in all aspects of the research process, including data analysis and report writing.
- Conducted synthesis of existing literature to understand current methods, possible avenues for improvements, and the potential use cases for our project.
- Ran trials to test performance metrics, including precision and recall, across different models and with the various dataset modifications outlined earlier in the paper.
- Conducted comprehensive reviews of proposal, progress, and final report before submissions. Provided feedback and revised shortcomings before deadlines.
- Created visualizations for the data and our analysis, which makes it much easier for fellow researchers to understand the results of our paper.
- Update and review Textdata posts regarding the project.
- Collected various possible datasets and generated report of pros/cons and possible problems to solve for each one.
- Created visualizations for the data and our analysis, which makes it much easier for fellow researchers to understand the results of our paper.
- Given 50% due to importance of role as leader and scope of work spanning across many areas of the project. On top of that, kept the focus of the group on the right topics to make sure tasks were completed in a smooth, focused, and timely fashion.

9.2 Joseph Thomas: Lead Developer - 50%

- Led implementation of research models and generated initial data to be analyzed. Implemented techniques for data augmentation and model training from reference papers that improved model robustness.
- Conducted research on possible problems to solve and providing context and theoretical grounding for the research findings.
- Explored existing literature to understand current methods, possible avenues for improvements, and the potential use cases for our project.
- Created code to apply previously outlined perturbations to images in order to train, test, and validate the models with this new data.

- Took algorithms from research papers and made improvements to increase precision and recall while reducing average loss.
- Managed the environment for code execution, ensured that the required libraries were installed correctly, and made sure the complete datasets were present on the machine. Also made sure the computational resources were sufficient to run data preprocessing and model training.
- Conducted comprehensive reviews of proposal, progress, and final report before submissions. Provided feedback and revised shortcomings before deadlines.
- Drafted significant portions of the results and discussion sections, articulating the implications of the findings on the field of adversarial machine learning.
- In charge of GitHub repository that includes updating the codes and files.
- Wrote extensive documentation of methodologies, research processeses, and results to support reproducibility and future audits.
- Illustrated model topology to allow for other researchers to create the models for further testing and research.
- Created visualizations for the data and our analysis, which makes it much easier for fellow researchers to understand the results of our paper.
- Conducted comprehensive reviews of proposal, midterm, and final report before submissions. Provided feedback and revised shortcomings before deadlines.
- Given 50% due to importance of work on the model to the project. Additionaly the code to test and generate data for each model presented interesting results to be analyzed.

10 REPRODUCIBILITY

Our research could be reproduced using this GitHub link: https://github.com/JosephPThomas/CS510_Team17/tree/main

REFERENCES

[1] Bernd Porr, Sama Daryanavard, L. M. B. H. C., and Dahiya, R. Real-time noise cancellation with deep learning.

[2] Devis Styo Nugroho, H. K., and Sardjono, T. A. Automatic sound alarm classification using deep learning for the deaf and hard of hearing.

[3] Jivitesh Sharma, O.-C. G., and Goodwin, M. Emergency detection with environment sound using deep convolutional neural networks. *SpringerLink 1184* (2021).