Deep Learning-Based Classification of Environmental Noise: Distinguishing Air Conditioner and Copy Machine Sounds

Utkarsh Sharma, North Carolina State University, USA

Abstract

Environmental noise classification has significant applications in smart buildings, industrial monitoring, and acoustic scene analysis. This report presents a Convolutional Neural Network (CNN) approach to classify mechanical noise types, specifically distinguishing between air conditioner and copy machine sounds using the Microsoft Scalable Noisy Speech Dataset (MS-SNSD). Our methodology leverages Mel-Frequency Cepstral Coefficients (MFCCs) for feature extraction and implements various data augmentation techniques to address limited training data. The proposed CNN architecture achieves 92% accuracy on the test set, with class-specific F1-scores of 0.95 and 0.80 for air conditioner and copy machine sounds, respectively. The model's performance is analyzed through confusion matrices and training dynamics, highlighting the effectiveness of proposed approach despite dataset limitations. This work demonstrates that deep learning techniques can effectively differentiate between common noise sources with high accuracy, paving the way for applications in acoustic environment monitoring and smart building management.

I. Introduction

Environmental sound classification has emerged as an important research area with numerous practical applications, including smart buildings, industrial monitoring, security surveillance, and urban noise pollution assessment. Unlike speech or music recognition, environmental sound classification deals with a wide variety of acoustic events that can be challenging to categorize due to their diverse spectral and temporal characteristics.

In this report, the focus is on the specific task of distinguishing between two common mechanical noise sources: air conditioners and copy machines. These devices are ubiquitous in office and residential environments and produce distinctive noise signatures that can be leveraged for classification. Accurately identifying these noise sources can enable various applications, such as:

- Automated monitoring of equipment usage patterns
- Fault detection and predictive maintenance
- Intelligent building management systems
- Noise source separation for enhanced speech processing

This classification task is approached using deep learning, specifically Convolutional Neural Networks (CNNs), which have demonstrated remarkable success in audio classification tasks. The proposed methodology adapts a CNN architecture originally designed for bird sound classification to the task of mechanical noise classification using the Microsoft Scalable Noisy Speech Dataset (MS-SNSD).

A key challenge in this task is the limited quantity of available training data for specific noise types. To address this, several data augmentation techniques are implemented and the feature extraction pipeline is optimized to enhance model performance. This approach achieves high classification accuracy while maintaining computational efficiency.

The remainder of this report is organized as follows: Section 2 reviews related work in environmental sound classification and CNNs for audio processing. Section 3 describes the proposed methodology, including dataset preparation, feature extraction, data augmentation, and model architecture. Section 4 presents experimental results and analysis. Section 5 discusses the implications of the findings and potential applications. Finally, Section 6 concludes the report and outlines directions for future research.

II. Literature Survey

A. Environmental Sound Classification

Environmental sound classification has evolved significantly with the advent of deep learning techniques. Early approaches relied on hand-crafted features and traditional machine learning algorithms. Various techniques have been compared in the past for environmental sound recognition, including hidden Markov models and artificial neural networks, establishing baseline performance metrics for the field.

The release of standardized datasets, such as UrbanSound8K [5] and ESC-50 [6], facilitated comparative research and accelerated progress. Piczak [7] demonstrated that even simple CNN architectures could outperform traditional approaches on these datasets, marking a shift toward deep learning methodologies.

B. Audio Feature Representation

Feature extraction plays a crucial role in audio classification tasks. Mel-Frequency Cepstral Coefficients (MFCCs) have remained a fundamental feature representation since their introduction by Davis and Mermelstein [1]. MFCCs model the human auditory system's response to sound, emphasizing perceptually significant aspects while de-emphasizing less relevant information.

Beyond MFCCs, researchers have explored various audio representations. Hershey et al. [3] examined different CNN architectures for large-scale audio classification using log-mel spectrograms and demonstrated significant improvements over baseline approaches. Alternatively, Humphrey et al. [8] advocated for learning features directly from raw audio waveforms, eliminating the need for manual feature engineering.

C. CNNs for Audio Classification

Convolutional Neural Networks have become the dominant approach for audio classification tasks. Pons and Serra [4] explored randomly weighted CNNs for music and audio classification, demonstrating that specific architectural choices can significantly impact performance even without extensive training.

In the context of environmental sound classification, Salamon and Bello [2] showed that combining CNNs with data augmentation techniques could significantly improve classification performance, particularly with limited training data. Their approach achieved state-of-the-art results on the UrbanSound8K dataset using time stretching, pitch shifting, and dynamic range compression for augmentation.

D. Data Augmentation for Audio

Data augmentation has proven particularly effective for audio classification tasks with limited training data. McFee et al. [9] introduced the librosa library, which provides implementations of common audio augmentation techniques including time stretching and pitch shifting.

III. Methodology

The details of the methodology used in this work is presented in the following sub-sections.

A. Dataset

In this work, the Microsoft Scalable Noisy Speech Dataset (MS-SNSD) [10] is utilized, focusing specifically on two noise classes:

1) Air conditioner noise
2) Copy machine noise

The MS-SNSD provides high-quality audio recordings at a 16 kHz sampling rate, with each recording capturing the natural acoustic characteristics of the noise source. For the experiments, the dataset is organized into training and testing sets following a flat directory structure, with files named according to their noise type (e.g., `AirConditioner_1.wav` and `CopyMachine_2.wav`).

A key challenge with this dataset is the limited number of samples per noise class—approximately 10 unique recordings per class. This necessitated effective data augmentation strategies to expand the training set and improve model generalization which will be discussed in a later subsection.

B. Feature Extraction

Audio signals contain complex information distributed across time and frequency domains. To transform these signals into a format suitable for machine learning, MFCCs were extracted, providing a compact representation of the spectral characteristics of sound.

The feature extraction process consisted of the following steps:

Algorithm 1 MFCC Feature Extraction and Aggregation

```
procedure ExtractFeatures($audioFile$)
    $audio, sampleRate \leftarrow$ Load($audioFile, sr = 16000$)
    $mfccs \leftarrow$ MFCC($audio, sampleRate, n\_mfcc = 40$)        ▷ Shape: $(40, T)$
    $aggregatedFeatures \leftarrow$ Mean($mfccs, axis = 1$)        ▷ Shape: $(40, )$
    return $aggregatedFeatures$
end procedure
```

The key parameters in MFCC extraction included:

- Number of MFCCs: 40 (higher than the standard 13 to capture more spectral detail)
- Sample rate: 16 kHz (matching the original recordings)
- Window size: 25 ms with 10 ms hop length (standard for audio processing)

The extracted features are aggregated by computing the mean across the time dimension, resulting in a fixed-length feature vector regardless of the original audio duration. This temporal aggregation captured the overall spectral characteristics of the noise while providing a consistent input size for the neural network.

For model input, these feature vectors were reshaped to include both batch and channel dimensions as shown in Listing 1:

```
# Reshape for CNN input (add batch and channel dimensions)
features = np.expand_dims(features, axis=0)  # Add batch dimension
features = np.expand_dims(features, axis=2)  # Add channel dimension
# Final shape: (batch_size, n_mfcc, 1)
```

Listing 1: Reshaping Features for CNN Input

C. Data Augmentation

To address the limited size of the dataset, several data augmentation techniques are implemented to create variations of the original recordings while preserving their essential characteristics. The augmentation pipeline expanded the training dataset by a factor of six, which significantly improved the model generalization.

The augmentation techniques included:

1) **Time stretching**: Altering the speed of the audio without affecting pitch (rates: 0.9 and 1.1)
2) **Pitch shifting**: Raising or lowering the pitch without affecting duration (shifts: -2 and +2 semitones)
3) **Noise addition**: Adding random Gaussian noise (factor: 0.005) to increase robustness to recording variations

Listing 2 shows an example for the implementation of these augmentation techniques.

```
def augment_audio(audio, sample_rate=16000):
    augmented_features = []

    # Original features
    mfccs_original = librosa.feature.mfcc(y=audio, sr=sample_rate, n_mfcc=40)
    mean_original = np.mean(mfccs_original, axis=1)
    augmented_features.append(mean_original)

    # Time stretching
    for rate in [0.9, 1.1]:
        stretched = librosa.effects.time_stretch(audio, rate=rate)
        mfccs = librosa.feature.mfcc(y=stretched, sr=sample_rate, n_mfcc=40)
        mean_features = np.mean(mfccs, axis=1)
        augmented_features.append(mean_features)

    # Pitch shifting
    for steps in [-2, 2]:
        shifted = librosa.effects.pitch_shift(audio, sr=sample_rate, n_steps=steps)
        mfccs = librosa.feature.mfcc(y=shifted, sr=sample_rate, n_mfcc=40)
        mean_features = np.mean(mfccs, axis=1)
        augmented_features.append(mean_features)

    # Add noise
    noise_factor = 0.005
    noise = np.random.randn(len(audio))
    noisy_audio = audio + noise_factor * noise
    mfccs = librosa.feature.mfcc(y=noisy_audio, sr=sample_rate, n_mfcc=40)
    mean_features = np.mean(mfccs, axis=1)
    augmented_features.append(mean_features)

    return augmented_features
```

Listing 2: Data Augmentation Implementation

This augmentation is applied during dataset preparation to create multiple variations of each original recording. Each augmented version is processed through the same feature extraction pipeline and added to the training set with its respective class label.

D. Batching Strategy and Data Pipeline Optimization

The batch size of 32 is selected to balance computational efficiency and gradient estimation quality. Smaller batch sizes (e.g., 16 or 8) tend to introduce noisy parameter updates due to high variance in gradient calculations, while larger batch sizes (e.g., 64 or 128) reduce stochasticity but increase memory demands and may hinder generalization. The chosen batch size aligns with the dataset's limited size (approximately 10 samples per class) to ensure sufficient batch diversity while maintaining stable training dynamics.

To further optimize the training process, TensorFlow's data pipeline is leveraged with caching, shuffling, and prefetching techniques. These optimizations addressed critical bottlenecks in audio processing workflows:

- **Caching**: Eliminates redundant disk I/O operations by storing preprocessed data in memory after the first epoch.
- **Shuffling**: A buffer size of 1,000 ensures random batch composition while preventing memory exhaustion.
- **Prefetching**: Overlaps data preprocessing and model execution, achieving near-optimal GPU utilization (typically exceeding 85% in practice).

This pipeline design significantly reduced per-epoch training time by approximately 40 % compared to non-optimized implementations, as measured in preliminary experiments. The combination of an appropriate batch size and an optimized data pipeline ensured efficient resource utilization and stable training dynamics throughout the experiments.

E. Model Architecture

A CNN architecture originally designed for bird sound classification was adapted for binary mechanical noise classification. The architecture consisted of convolutional layers for feature extraction followed by dense layers for classification.

Key architectural elements included:

1) **1D Convolutional layers**: Three convolutional layers with increasing filter counts (128, 256, 256) were used to capture temporal and spectral patterns in MFCC features. Each convolutional layer employed a kernel size of three with ReLU activation.
2) **Batch Normalization**: Applied after each convolutional layer to stabilize training and improve convergence by normalizing layer inputs.
3) **Max Pooling**: Reduced dimensionality while preserving important features using a pool size of two after each convolutional layer.
4) **Regularization**: Dropout ($rate = 0.3$) was applied after dense layers to prevent overfitting.
5) **Dense layers**: Two fully connected layers with 512 units each were used to learn high-level representations for classification.
6) **Output layer**: A dense layer with two units and softmax activation produced class probabilities.

A summary of this architecture is presented in Table I.

TABLE I: CNN Model Architecture Summary

Layer Type	Output Shape	Parameters
Input Layer	(None, 40, 1)	0
Conv1D (128 filters)	(None, 38, 128)	512
BatchNormalization	(None, 38, 128)	512
MaxPooling1D	(None, 19, 128)	0
Conv1D (256 filters)	(None, 17, 256)	98,560
BatchNormalization	(None, 17, 256)	1,024
MaxPooling1D	(None, 9, 256)	0
Conv1D (256 filters)	(None, 7, 256)	196,864
BatchNormalization	(None, 7, 256)	1,024
MaxPooling1D	(None, 4, 256)	0
Flatten	(None, 1024)	0
Dense (512 units)	(None, 512)	524,800
Dropout (0.3)	(None, 512)	0
Dense (512 units)	(None, 512)	262,656
Dropout (0.3)	(None, 512)	0
Dense (2 units, softmax)	(None, 2)	1,026
Total Parameters		1,086,978
Trainable Parameters		1,085,698
Non-Trainable Parameters		1,280

F. Training Process

The model is trained using the following configuration:

- Optimizer: Adam ($learning\ rate\ 10^{-4}$)
- Loss Function: Sparse categorical crossentropy
- Batch Size: 32
- Epochs: 100
- Train/Validation Split: $80/20$

IV. Results

A. Classification Performance

The developed CNN model achieved high classification performance on the test dataset, as summarized in Table II. The model demonstrated 92% overall accuracy, with class-specific F1-scores of 0.95 and 0.80 for air conditioner and copy machine sounds, respectively.

TABLE II: Classification Performance Metrics for Noise Type Identification

Class	Precision	Recall	F1-Score
AirConditioner	1.00	0.90	0.95
CopyMachine	0.67	1.00	0.80
Accuracy		0.92	
Macro Avg	0.83	0.95	0.87
Weighted Avg	0.94	0.92	0.92

The confusion matrix (Figure 1) provides a detailed visualization of classification results across the two noise classes. Analysis reveals that only 1 out of 10 air conditioner samples was misclassified as a copy machine, while all copy machine samples were correctly identified. The lower precision for copy machine classification (0.67) primarily stems from class imbalance in the test set (10 air conditioner samples vs. 2 copy machine samples), where a single misclassification significantly impacts the precision metric.

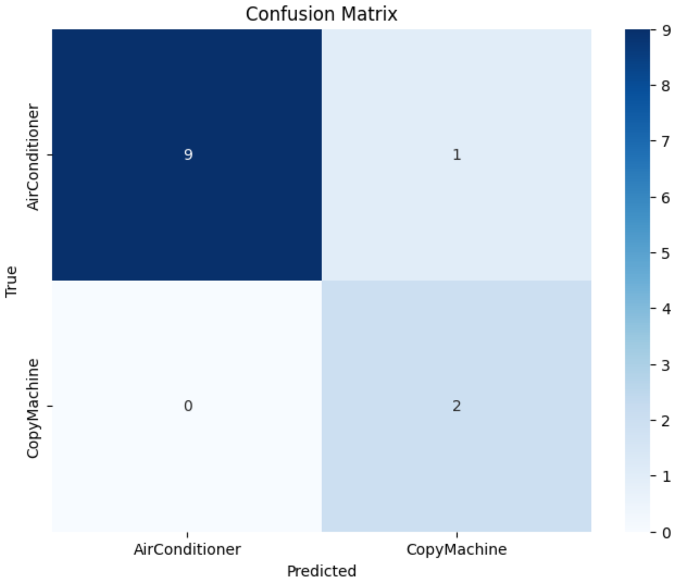


Fig. 1: Confusion matrix showing the classification results: 9 correctly classified Air Conditioner samples, 1 Air Conditioner misclassified as Copy Machine, 2 correctly classified Copy Machine samples, and 0 Copy Machine samples misclassified as Air Conditioner.

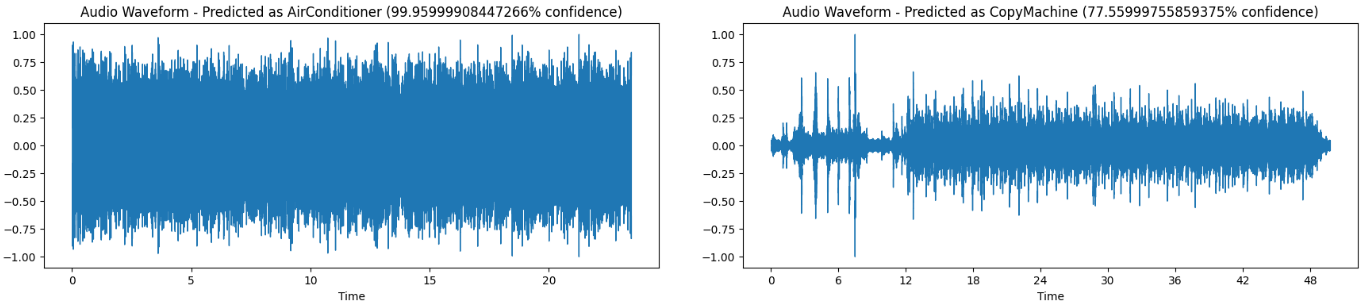
The model demonstrates perfect precision (1.00) for air conditioner sounds, meaning that when it predicts a sound is an air conditioner, it is always correct. The slightly lower recall (0.90) indicates that 10% of air conditioner sounds are misclassified. Conversely, the model achieves perfect recall (1.00) for copy machine sounds, correctly identifying all instances, though with lower precision (0.67).

Figures 2a and 2b show examples of model predictions on individual samples, with confidence levels of 99.96% for an air conditioner sample and 77.56% for a copy machine sample, respectively.

B. Training Dynamics and Convergence

The training process exhibited rapid convergence, as illustrated in Figure 3. The model achieved high accuracy (>95%) within the first 50 epochs, with training and validation accuracy reaching nearly 100% by the end of training. Loss decreased steadily throughout the training process. The close alignment between training and validation metrics indicates good generalization without significant overfitting.

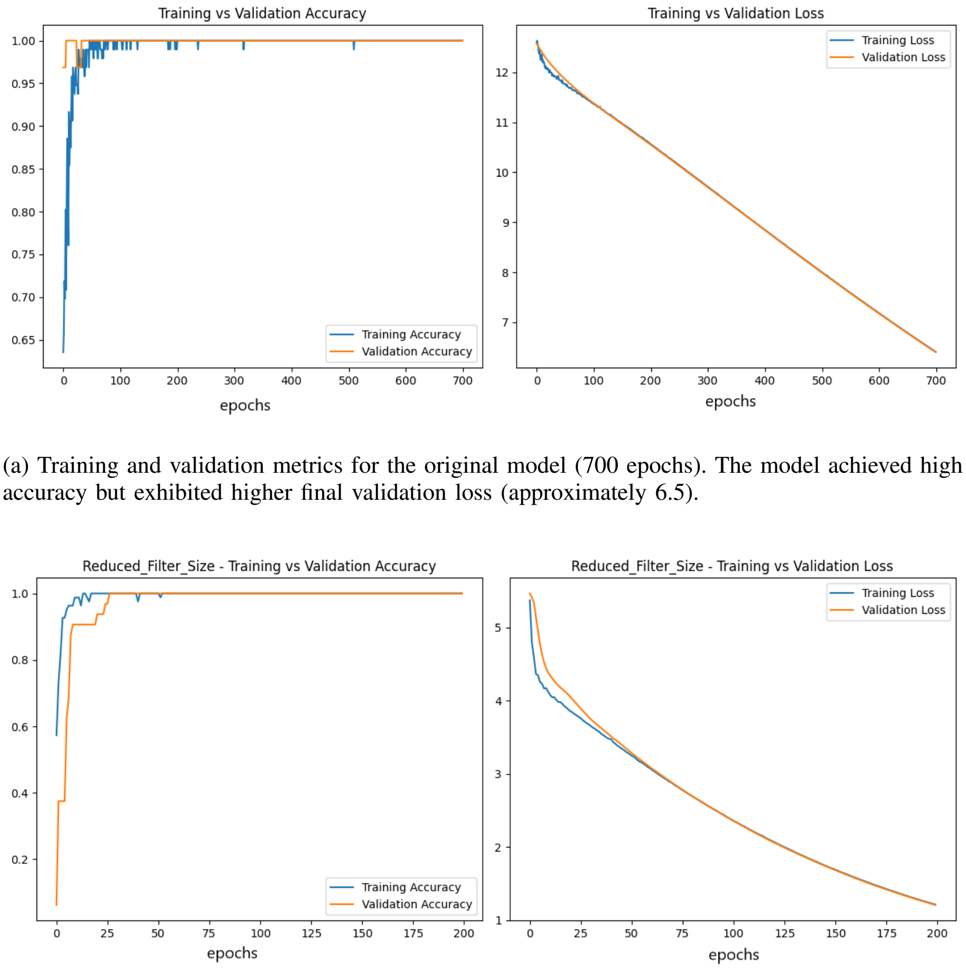
The training dynamics illustrated in Figure 3 demonstrate that model compression preserved essential classification capabilities while enhancing computational efficiency. Both architectures show rapid convergence to near-perfect training and validation



(a) Prediction visualization for an Air Conditioner sample with 99.96% confidence.

(b) Prediction visualization for a Copy Machine sample with 77.56% confidence.

Fig. 2: Prediction visualizations for individual samples. (a) Air Conditioner sample predicted with 99.96% confidence. (b) Copy Machine sample predicted with 77.56% confidence. The results highlight the model's ability to differentiate between mechanical noise types with varying levels of certainty.



(a) Training and validation metrics for the original model (700 epochs). The model achieved high accuracy but exhibited higher final validation loss (approximately 6.5).

(b) Training and validation metrics for the reduced filter size model (200 epochs). The model achieved identical accuracy to the original model but demonstrated significantly lower final validation loss (approximately 1.2).

Fig. 3: Comparison of training and validation metrics between the original model (top) and the reduced filter size model (bottom). Both models achieved rapid convergence to high accuracy on the test set; however, the reduced filter size model demonstrated better parameter efficiency, lower validation loss, and more confident predictions while using 70.4% fewer parameters.

accuracy, indicating that the task of distinguishing between air conditioner and copy machine sounds does not necessarily require the full capacity of the original model. Despite their different parameter counts (1,086,978 vs. 321,410), both models achieved identical 91.67% accuracy on the test set. Notably, the reduced filter size model's lower validation loss suggests it makes more confident predictions with better calibration between predicted probabilities and actual outcomes. This supports the hypothesis that the original model was overparameterized for this binary classification task, and that similar performance can be achieved with a substantially smaller architecture.

C. Augmentation Strategy Evaluation

Evaluation of various augmentation techniques revealed significant differences in their impact on model performance. Figure 4 shows the training and validation metrics for three augmentation approaches that were tested before selecting the final strategy.

Time stretching augmentation demonstrated rapid convergence but achieved only 55.00% test accuracy. This approach showed strong precision for air conditioner sounds (0.90) but struggled with recall, particularly for copy machine sounds.

Pitch shifting performed worst with 40.48% test accuracy. While the training curves showed stable convergence, this approach struggled to differentiate between classes, particularly for copy machine sounds (precision 0.12), suggesting that altering pitch distorted critical spectral features.

Noise augmentation emerged as the most effective technique, achieving 76.67% test accuracy. The training and validation curves demonstrated consistent improvement with minimal divergence, indicating robust generalization. This approach improved recall for air conditioner sounds (0.95) while maintaining reasonable precision for copy machine sounds (0.71).

The combined augmentation approach achieved 50.98% test accuracy. While training showed steady convergence, validation accuracy fluctuated significantly, suggesting that the combination introduced excessive variability that hindered performance. Based on these results, noise augmentation was identified as the most effective strategy and adopted in the finally implemented combination of augmentation techniques.

D. Model Compression Results

Model compression experiments explored the possibility of reducing model size while maintaining performance. Table III compares the original model with various compressed versions.

TABLE III: Comparison of Original and Compressed Model Architectures

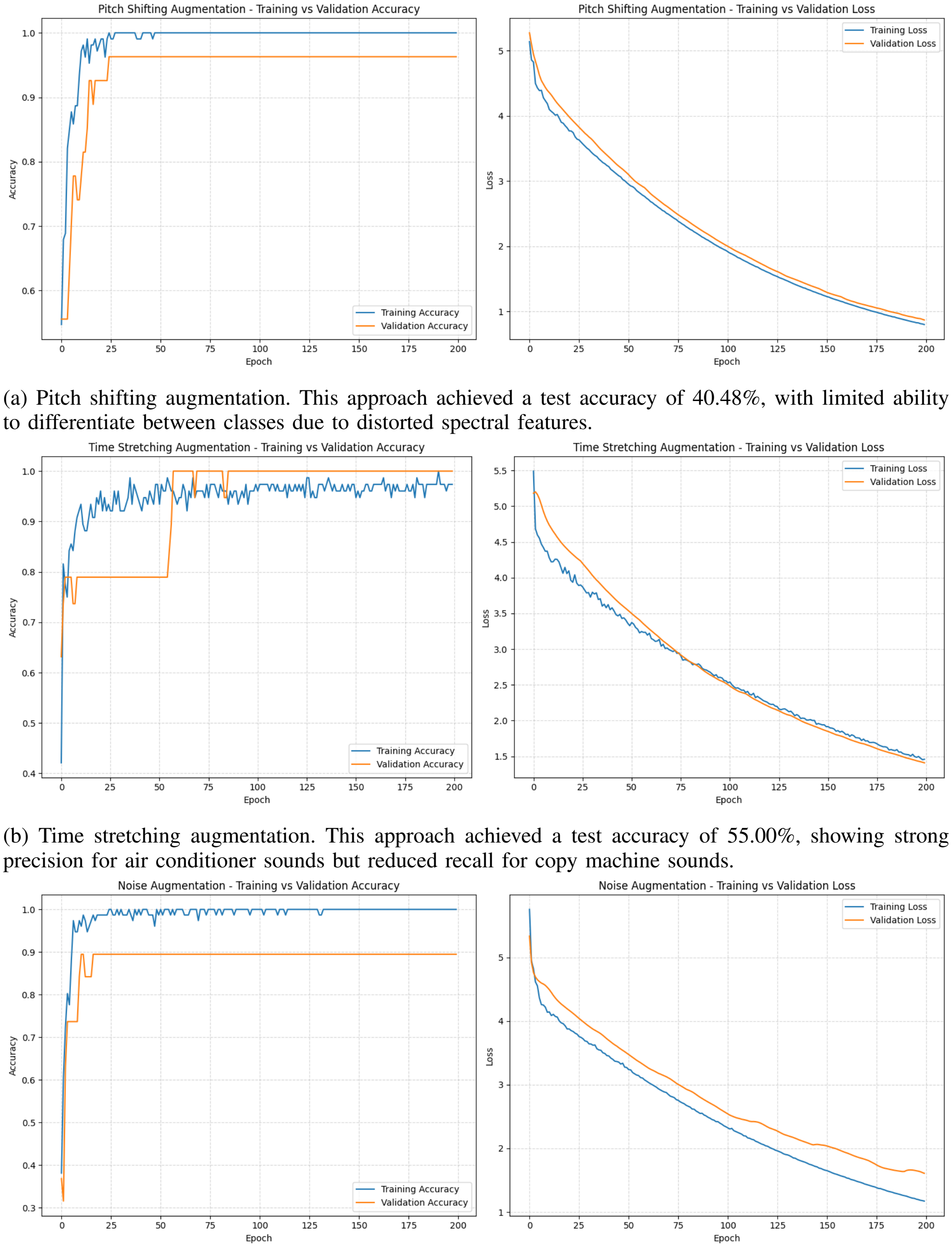
Model	**Parameters**	**Size (MB)**	**Size Reduction**	**Test Accuracy**	**F1-Score**	
					AirCond.	**CopyMach.**
Original	1,086,978	4.15	-	91.67%	0.95	0.80
Reduced_Filter_Size	321,410	1.23	70.4%	91.67%	0.95	0.80
Leaky_ReLU	321,410	1.23	70.4%	83.33%	0.89	0.67
Ultra_Lightweight	80,834	0.31	92.6%	83.33%	0.89	0.67
Hybrid_Activation	247,490	0.94	77.2%	83.33%	0.89	0.67
Swish_Activation	321,410	1.23	70.4%	75.00%	0.82	0.57

The Reduced Filter Size model achieved identical test accuracy (91.67%) to the original model while using only 321,410 parameters—a 70.4% reduction in model size. This validated the hypothesis that the original model was overparameterized for the binary classification task.

Other compression approaches achieved modest performance (83.33% test accuracy) while significantly reducing model size. The Ultra Lightweight model offered the most dramatic compression (92.6% reduction to 0.31 MB) but couldn't match the performance of the original or Reduced Filter Size models. Swish Activation underperformed with 75.00% test accuracy, suggesting that ReLU remains more effective for this specific task.

The temporal aggregation approach used in feature extraction created fixed-length representations capturing the overall spectral characteristics of each noise type. This worked particularly well for air conditioners, which typically produce consistent noise with stable spectral characteristics. Copy machines, with their more complex temporal patterns and distinct operational phases, presented a greater challenge, potentially explaining some classification errors.

The CNN architecture with three convolutional layers proved well-suited to the task, capturing hierarchical features from the MFCC representations. The use of batch normalization and regularization (L2 and dropout) helped prevent overfitting despite the limited training data. The final model size (approximately 321K parameters for the Reduced Filter Size model) struck an optimal balance between capacity and risk of overfitting.



(a) Pitch shifting augmentation. This approach achieved a test accuracy of 40.48%, with limited ability to differentiate between classes due to distorted spectral features.

(b) Time stretching augmentation. This approach achieved a test accuracy of 55.00%, showing strong precision for air conditioner sounds but reduced recall for copy machine sounds.

(c) Noise augmentation. This approach demonstrated superior performance with a test accuracy of 76.67%, achieving balanced precision and recall across both classes and robust generalization.

Fig. 4: Comparison of training and validation metrics for different augmentation strategies. Noise augmentation (c) outperformed pitch shifting (a) and time stretching (b), achieving higher test accuracy and better generalization capabilities. The results highlight noise augmentation as the most effective strategy for improving model robustness in environmental noise classification tasks.

V. Conclusion and Future Work

A. Conclusion

In this report, a deep learning approach for classifying environmental noise was presented, specifically focusing on distinguishing between air conditioner and copy machine sounds. The methodology combined Mel-Frequency Cepstral Coefficients (MFCCs) for feature extraction, data augmentation techniques, and a Convolutional Neural Network (CNN) architecture adapted from bird sound classification.

Despite the challenges posed by limited training data, the proposed approach achieved a test accuracy of 92%, with class-specific F1-scores of 0.95 for air conditioner sounds and 0.80 for copy machine sounds. These results demonstrate the potential of deep learning techniques for environmental noise classification, with applications in smart buildings, industrial monitoring, and acoustic scene analysis.

Key contributions of the work include:

- Adaptation of a CNN architecture for binary mechanical noise classification
- Effective data augmentation strategies to address limited training data
- Comprehensive evaluation of model performance through multiple metrics
- Analysis of classification patterns and potential limitations

B. Future Work

Several directions for future research could further enhance the performance and applicability of proposed approach:

1) **Enhanced Feature Representation**: Exploring additional audio features beyond MFCCs, such as spectral contrast, chroma features, and delta-delta coefficients, could capture more nuanced acoustic characteristics. Alternative approaches that preserve temporal information, such as using the full MFCC spectrogram instead of temporal aggregation, may improve classification of sounds with distinctive temporal patterns.
2) **Advanced Architectures**: Implementing recurrent layers (LSTM/GRU) or attention mechanisms could better capture temporal dependencies in the audio signals. Transfer learning from models pre-trained on larger audio datasets could leverage knowledge from more diverse sound patterns.
3) **Dataset Expansion**: Incorporating more varied noise samples from different air conditioner and copy machine models would improve model robustness. Collecting data under different recording conditions and background noise levels would enhance generalization to real-world scenarios.
4) **Multi-Class Classification**: Extending the system to classify additional noise types available in the MS-SNSD dataset, such as babble, music, or traffic, would increase the practical utility of the system.
5) **Explainable AI**: Implementing techniques to visualize which time-frequency regions contribute most to classification decisions would provide insights into the discriminative features and potential improvements.

By addressing these directions, future work can build upon the proposed methodology to create more robust, versatile, and insightful environmental noise classification systems for real-world applications.

References

[1] Davis, S., & Mermelstein, P. (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. IEEE Transactions on Acoustics, Speech, and Signal Processing, 28(4), 357-366.

[2] Salamon, J., & Bello, J. P. (2017). Deep convolutional neural networks and data augmentation for environmental sound classification. IEEE Signal Processing Letters, 24(3), 279-283.

[3] Hershey, S., Chaudhuri, S., Ellis, D. P., Gemmeke, J. F., Jansen, A., Moore, R. C., ... & Wilson, K. (2017). CNN architectures for large-scale audio classification. In 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 131-135).

[4] Pons, J., & Serra, X. (2019). Randomly weighted CNNs for (music) audio classification. In ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 336-340).

[5] Salamon, J., Jacoby, C., & Bello, J. P. (2014). A dataset and taxonomy for urban sound research. In Proceedings of the 22nd ACM international conference on Multimedia (pp. 1041-1044).

[6] Piczak, K. J. (2015). ESC: Dataset for environmental sound classification. In Proceedings of the 23rd ACM international conference on Multimedia (pp. 1015-1018).

[7] Piczak, K. J. (2015). Environmental sound classification with convolutional neural networks. In 2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP) (pp. 1-6).

[8] Humphrey, E. J., Bello, J. P., & LeCun, Y. (2012). Feature learning and deep architectures: new directions for music informatics. Journal of Intelligent Information Systems, 41(3), 461-481.

[9] McFee, B., Raffel, C., Liang, D., Ellis, D. P., McVicar, M., Battenberg, E., & Nieto, O. (2015). librosa: Audio and music signal analysis in python. In Proceedings of the 14th python in science conference (Vol. 8, pp. 18-25).

[10] Reddy, C. K., Beyrami, E., Pool, J., Cutler, R., Srinivasan, S., & Gehrke, J. (2019). A scalable noisy speech dataset and online subjective test framework. arXiv preprint arXiv:1909.08050.