Exploring CNN, LSTM, and Attention Mechanism with Augmentation Techniques on ESC-50 Dataset

Roben Bhatti†

***Abstract*—This paper explores the performance of three different deep learning models: CNN, CNN-LSTM and CNN-LSTM-Attention for environmental sound classification using the ESC-50 dataset. The dataset, consisting of 2000 labeled audio samples from 50 sound classes, poses challenges due to its limited size. To improve classification accuracy, the study investigates the impact of both audio and image-based data augmentation techniques. The models were tested across multiple data augmentation scenarios, and the performance was evaluated using a cross-validation approach to ensure robustness.**

Results show that audio augmentation consistently improves model performance, leading to significant accuracy gains across all models. Conversely, image augmentation negatively impact performance. Among the tested models, the addition of a simple attention mechanism increased accuracy by 2%, with only a modest increase of 200 parameters. The model that reach the highest validation accuracy (75%) is the CNN trained with audio augmentations, the highest accuracy along the models trained with the original dataset is of 61% and it belongs to the CNN-LSTM-Attention model. However, challenges remain in optimizing these models given the limited dataset size, and future work could explore more sophisticated attention models to further enhance performance as SOTA models. The findings emphasize the necessity of effective data augmentation and targeted architectural improvements when working with small audio datasets.

***Index Terms*—Environmental sound classification, Supervised Learning, Neural Networks, Recurrent Neural Networks, Attention mechanism, Data augmentation**

I. Introduction

Sound classification is a well known task in various domains, including environmental monitoring and human-computer interaction. Classification of environmental sounds can enhance the functionality of systems in fields like smart cities [1] and autonomous vehicles [2], making them more aware of and responsive to their surroundings. This paper focuses on exploring environmental sound classification using the ESC-50 dataset [3], a benchmark dataset widely used for this task, which presents unique challenges due to its diverse sound categories and complex audio patterns.

Over the years, numerous approaches have been tested on the dataset, but only a few have managed to match or surpass the estimated human classification accuracy of 81.30%.

In this work, we implement and evaluate three models: a Convolutional Neural Network (CNN), CNN with Long Short-Term Memory (LSTM) layers, and CNN with LSTM plus an Attention mechanism. These models are tested on the full dataset using cross-validation. We further explore three approaches to different preprocessings: using normal Mel Spectrograms, applying audio augmentation before generating the spectrogram, and leveraging image augmentation on the spectrogram. Our results demonstrate how each combination impacts performance, providing insights into the best strategies for robust environmental sound classification.

- We compare three deep learning models: CNN, CNN + LSTM, and CNN + LSTM + Attention using the ESC-50 dataset.
- We explore and evaluate the effects of three feature extraction methods: standard mel spectrograms, audio augmentation followed by mel spectrograms, and image augmentation applied to mel spectrograms.
- Our findings reveal the optimal combinations of models and augmentation techniques for improving classification accuracy and robustness in environmental sound classification tasks
- The proposed methods offer practical insights for deploying sound classification systems in real-time applications, such as autonomous vehicles and IoT-based monitoring systems.

The paper is structured as follows: after reviewing related work in Section II, we describe the signals and features used for classification in Section IV, which includes details on the dataset, cross-validation process, the generation of mel spectrograms, and the augmentation techniques employed. Section V focuses on the deep learning models applied in our experiments, while Section VI presents the results of the different models on the ESC-50 dataset. Finally, the findings are discussed, with an emphasis on how various models and augmentation strategies impact classification performance.

II. Related Work

The release of the ESC-50 dataset by Karol Piczak in 2015 provided a significant boost to this research area. ESC-50 contains 2,000 short audio clips spanning 50 different environmental sound classes, and it quickly became a benchmark for evaluating sound classification methods .

Piczak's initial work [3] demonstrated the potential of traditional machine learning models, including feature extraction-based classifiers, for environmental sound classification. These models were compared against human performance on the ESC-50 dataset, providing a baseline for future developments . However, subsequent works [4] marked a shift towards deep learning methods, particularly convolutional neural networks (CNNs), applied to spectrograms derived from audio clips.

† Physics of Data, Department of Physics and Astronomy, University of Padova, email: roben.bhatti@studenti.unipd.it

Since then, the ESC-50 dataset has served as a competitive platform for researchers to test novel approaches. Over 30 papers have been published, presenting a wide range of classification techniques, many of which are cataloged on the dataset's official GitHub page. Among the most successful methods are CNN-based architectures, which consistently outperform other models, including those based on Mel-frequency cepstral coefficients (MFCC) and recurrent neural networks (RNNs) . Various adaptations of CNNs, involving advanced preprocessing, data augmentation, and feature representations, have achieved the highest accuracies. For instance, several studies have implemented novel forms of spectrogram transformation [5] or explored deeper and more complex CNN structures, which surpass the accuracy of earlier models . In the last years, however, popularity of multimodal approaches in application to audio-related tasks has been increasing [6]. Another notable trend among top-performing models is the use of pretrained networks and data augmentation techniques to enhance model performance. These models often involve hybrid architectures that combine CNNs with other machine learning approaches such as attention mechanisms and residual connections. Despite the complexity of these models, they have proven to be highly effective, with the top three models reaching an impressive accuracy of up to 98% [7]–[9], significantly outperforming human classification accuracy on ESC-50 (81.30%) .

Overall, the landscape of environmental sound classification has shifted toward increasingly complex and effective CNN architectures, often supplemented with data augmentation and attention mechanisms. This work builds on these advancements by exploring novel combinations of CNNs with LSTM and Attention layers, alongside various augmentation techniques, to further improve classification accuracy on the ESC-50 dataset.

III. Processing Pipeline

The initial stage of the pipeline entails feature extraction, whereby the raw audio files are transformed into mel spectrograms, which offer a time-frequency representation of the sound. Mel spectrograms are particularly effective for capturing the spectral and temporal properties of audio signals, rendering them optimal inputs for deep learning models. In addition to utilising conventional mel spectrograms, two augmentation strategies are employed to enhance model generalisation: audio augmentation, which incorporates transformations such as pitch shifting and time stretching prior to generating spectrograms, and image augmentation, which modifies the spectrograms directly to augment data diversity.

Subsequently, the generated spectrograms are fed into one of three model architectures: a Convolutional Neural Network (CNN), a CNN combined with Long Short-Term Memory (LSTM) layers, or a CNN with LSTM and an attention mechanism. The convolutional neural network (CNN) is responsible for learning the spatial patterns within the spectrograms, effectively treating them as images and capturing the local features that are crucial for classification. In the second model, Long Short-Term Memory (LSTM) layers are incorporated to further capture temporal dependencies, thereby ensuring that the model takes into account the sequential nature of sound events. The third model incorporates an attention mechanism, which enables the system to focus on the most pertinent aspects of the feature vector.

The models are trained using cross-validation on the full ESC-50 dataset, thereby ensuring a robust evaluation of performance. By comparing the three models across different feature extraction and augmentation strategies, insights can be gained into the optimal combinations that lead to better classification accuracy. Each model is trained to predict one of the 50 sound categories, and their outputs are compared in order to determine which architecture and processing pipeline yield the best results.

IV. Signals and Features

A. ESC-50 Dataset

The ESC-50 dataset is a curated collection of 2,000 labeled environmental audio recordings designed for benchmarking environmental sound classification methods. Each recording is 5 seconds long and categorized into 50 distinct semantic classes, with 40 examples per class. These classes are organized into five broad categories: animals, natural soundscapes and water, human non-speech, interior/domestic sounds, and exterior/urban noises. The recordings were manually selected from public field recordings provided by the Freesound.org project [10].

B. Mel spectrograms

To prepare the audio data, we begin by converting the raw audio signals into mel spectrograms using a short-time Fourier transform (STFT). The STFT eq. [1] operates on overlapping windows with a window length of 2048 samples and a stride of 512 samples. We apply a Hann windowing function to each segment, which helps reduce spectral leakage.

$$X(m,\omega) = \sum_{n=-\infty}^{\infty} x[n] * w[n-m] * e^{-j\omega n} \tag{1}$$

Where $x[n]$ is the input signal and $w[n-m]$ is the Hann Window centered in m, ω is the angular frequency of the fourier transform. The resulting magnitude spectrogram is $|X(m,\omega)|^2$ which is then transformed into a mel spectrogram with 128 mel bins, covering the frequency range from 20 Hz to 16,000 Hz. The spectrogram is then normalized to a decibel scale (dB), with a dynamic range capped at 80 dB to improve robustness to outliers and noise. Applying Mel filters fig. 1 (i.e., Mel spectrograms) provides a more accurate representation of the audio signal that aligns with human auditory perception. This makes Mel spectrograms more effective for tasks like speech and environmental sound classification, as they capture features that are more relevant to human listeners.

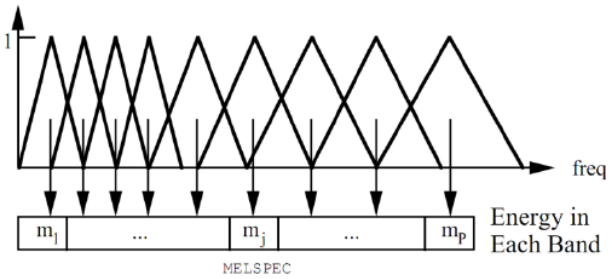


Fig. 1: Mel filterbank visualization: The filterbank captures finer details in the lower frequency range, while using fewer filters at higher frequencies to account for the human ear's logarithmic perception of sound.

C. Audio Augmentation

To further improve model generalization, we apply a series of audio augmentation techniques to the input audio. These augmentations help simulate variations in environmental conditions and reduce the risk of overfitting. The augmentations include:

- **Gaussian Noise**: Randomly adds noise with a minimum amplitude of 0.001 and a maximum amplitude of 0.015.
- **Time Stretching**: Randomly speeds up or slows down the audio by a factor ranging from 0.8 to 1.25.
- **Pitch Shifting**: Shifts the pitch of the audio by up to 4 semitones in either direction.
- **Shifting**: Applies random temporal shifts to the audio with a maximum shift of 0.5 seconds.

These transformations are applied with a probability of 0.5, ensuring that the augmented dataset retains sufficient variability to enhance model training.

D. Image augmentation

Once the audio data is preprocessed, mel spectrograms are computed. Each mel spectrogram is represented as a matrix of shape (128, T), where 128 corresponds to the number of mel frequency bins, and T is the number of time frames, determined by the length of the audio and the hop size used in the STFT.

In parallel to audio augmentation, we also apply image augmentation to the spectrograms, treating them as images. These augmentations are designed to simulate variability in the spectrogram domain and include:

- **Random Horizontal Flip**: Randomly flips the spectrogram along the time axis.
- **Random Brightness Adjustment**: Adjusts the brightness of the spectrogram image by up to 10%.
- **Random Translation**: Translates the spectrogram image by up to 10% in both the time and frequency axes.
- **Random Contrast Adjustment**: Alters the contrast of the spectrogram by a factor between 0.6 and 1.4.

These image augmentations are applied during the training phase to further improve model robustness. Figure **??** adresses the different preprocessing involved.

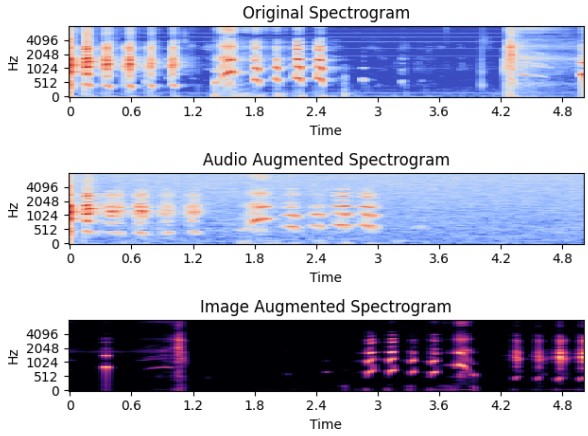


Fig. 2: upper figure: original Mel spectrogram, middle figure: Mel Spectrogram generated after audio augmentations, bottom figure: image augmentation on the original spectrogram. It is evident that in the bottom figure the spectrogram has been horizontally flipped.

E. Dataset Splitting

The dataset is split into training and validation sets using a cross-validation approach. Cross-validation ensures that each sample from the ESC-50 dataset is used for both training and evaluation across different folds, mitigating potential biases introduced by specific data partitions. This method provides a more reliable estimate of model performance. Each fold is evaluated independently, and the final performance is averaged across all folds.

Each fold consists of 400 samples in the validation set, while the training set includes the remaining 1,600 samples from the ESC-50 dataset. In cases where data augmentation is applied, the training set is expanded four times, resulting in a total of 6,400 samples, while the validation set remains untouched at 400 samples.

In particular for the audio augmentations the dataset is generated before the actual training and locally saved, instead the image augmentations are done on the fly during the training and on the original dataset (resampled at 16 KHz).

No augmentation is applied to the validation set to ensure that the evaluation process remains unbiased. Augmenting the validation set could artificially inflate performance metrics by introducing variations not present in the original data. By keeping the validation set unchanged, we guarantee that model evaluation reflects its ability to generalize to real, unseen data, rather than overfitting to augmented samples.

V. Learning Framework

In this section, we describe the learning framework and the approach used to tackle the environmental sound classification task with the ESC-50 dataset. Our framework is built upon convolutional neural networks (CNNs) for extracting meaningful features from spectrograms, and it incorporates techniques like data normalization, dropout regularization, and early stopping to ensure efficient learning and generalization.

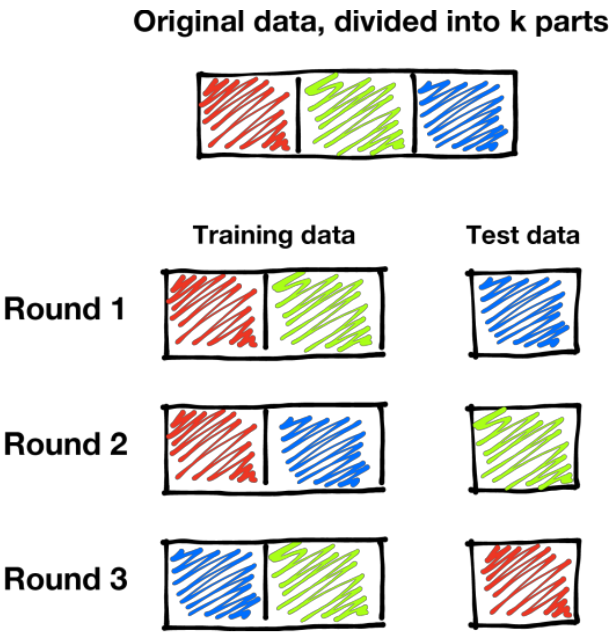


Fig. 3: Cross-validation process: The dataset is divided into multiple folds, where each fold is used as a validation set once while the remaining folds serve as the training set.

The architecture and training strategies were carefully chosen to balance model complexity and performance.

A. Training Strategy

1) Optimizer: The model is trained using the Adam optimizer [11], which provides efficient gradient-based optimization with adaptive learning rates. We use the Sparse Categorical Crossentropy loss function with logits to handle the multi label classification task.

2) Regularization and Early Stopping: To prevent overfitting and improve generalization, we incorporate several regularization techniques. The model is trained with dropout in both the convolutional and fully connected layers. Additionally, we use an early stopping mechanism that monitors the validation accuracy. If no improvement is observed for 15 consecutive epochs, training is halted to prevent unnecessary computation and to avoid overfitting.

3) Model Checkpointing: During training, we also employ a model checkpoint callback to save the model with the best validation accuracy. This ensures that the best-performing model is retained even if subsequent epochs lead to overfitting.

4) Normalization: Normalization layer is applied as a preprocessing keras layer step to ensure consistent scaling of input data across all models. Specifically, the normalization is performed using the standard scaler approach, which adjusts the input features by subtracting the mean and dividing by the variance calculated from the training data. This process standardizes the data, bringing it to a mean of zero and a variance of one. Such normalization is crucial in deep learning, particularly for neural networks, as it ensures that all input features are on a comparable scale, preventing certain features from dominating due to larger ranges.

B. CNN model

Our core model is a Convolutional Neural Network (CNN), fig. 4, designed to process 2-D mel spectrogram inputs, which are 120x120 in size after the resizing layer. The network consists of multiple convolutional layers interspersed with pooling layers and dropout for regularization. The layers are designed to progressively capture hierarchical features from the input spectrograms, moving from low-level frequency patterns to higher-level representations that can aid classification.

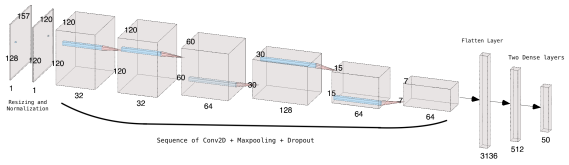


Fig. 4: CNN architecture: The tensor shapes refer to the output from the respective layer. Dropout and BatchNormalization are not represented in the picture for simplicity of visualization.

The CNN architecture includes:

- **Convolutional Layers**: The model features several convolutional layers with increasing numbers of filters (32, 64, 128) and 3x3 kernels. Each layer applies the ReLU activation function to introduce non-linearity and uses 'same' padding to preserve spatial dimensions.
- **Pooling Layers**: MaxPooling is applied after pairs of convolutional layers to downsample the feature maps, reducing the spatial resolution while retaining important features.
- **Dropout and Batch Normalization**: Dropout layers with a dropout rate of 0.2 and 0.5 are placed throughout the network to prevent overfitting by randomly deactivating a fraction of neurons during training. Batch Normalization is employed after deeper convolutional layers to stabilize learning by normalizing activations.

The final layers of the network are fully connected, leading to a dense layer of 512 neurons followed by another dropout layer. The output layer consists of 50 neurons corresponding to the 50 sound classes, with no activation function since we use a softmax-based loss during training.

C. CNN-LSTM

Building upon the CNN model architecture detailed in Section V-B, which serves as a feature extractor by learning local patterns in the mel spectrograms, the output from the CNN layers is reshaped and passed to the LSTM network. The key idea behind this hybrid model in fig. 5 is to leverage the CNN's ability to detect spatial features, such as frequency and energy patterns, while the LSTM layers focus on the temporal dependencies that exist within the audio signals.

In particular, after the convolutional layers, the output is reshaped using a permutation and flattening step, where the time dimension is moved to the forefront. This restructuring

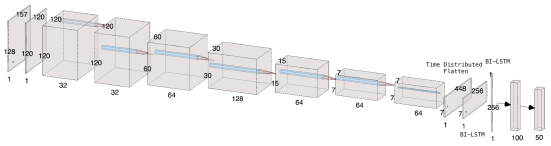


Fig. 5: CNN-LSTM architecture: The tensor shapes refer to the output from the respective layer.

allows the sequential LSTM layers to process the temporal relationships between different time frames. The first LSTM layer is a bidirectional LSTM with 128 units, which reads the data both forward and backward in time, thus capturing both past and future context in the audio. This is crucial for environmental sounds that have non-stationary patterns, where the presence of certain sounds at different times affects classification and also possible because we don't need real time capabilities for this analysis.

Following the first LSTM layer, a second bidirectional LSTM with another 128 units processes the data, further enhancing the model's ability to understand complex time dependencies. The LSTMs are equipped with dropout regularization (set at 0.25) to mitigate overfitting by randomly deactivating a fraction of neurons during training, thus improving the generalization of the model.

The dense layers following the LSTM layers are responsible for the final classification. A fully connected layer with 100 units is applied before the final output layer, which consists of 50 units, one for each of the ESC-50 classes. The final classification is computed using a softmax activation function to assign probabilities to each class.

D. CNN-LSTM-Attention

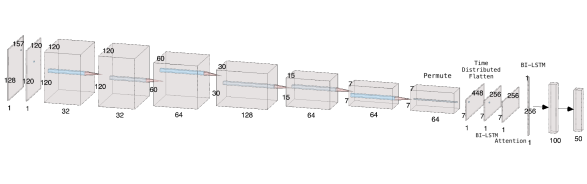


Fig. 6: CNN-LSTM-Attention architecture: The tensor shapes refer to the output from the respective layer.

The final model in fig. 6 extends the CNN-LSTM architecture by introducing a simple attention mechanism between the two LSTM layers to further enhance the temporal modeling of the input data. This attention-based hybrid architecture was inspired by Raffel et al. [12] work where they demonstrate that the resulting model can solve the synthetic "addition" and "multiplication" long term memory problems.

As in the previous model described in Section V-C, the feature extraction is handled by the CNN layers, which process the mel spectrograms and produce spatial feature maps. These maps are then reshaped and passed to a bidirectional LSTM layer with 128 units, which processes the data in both forward and backward directions to capture bidirectional temporal dependencies.

Following this first LSTM layer, an attention mechanism is applied. The attention layer enables the model to dynamically weigh the importance of different time steps within the sequence. Instead of treating all time steps equally, the attention layer allows the model to focus more on the critical moments in the sequence, where key features for sound classification might be present.

The attention mechanism eq.[2] operates by first calculating an importance score for each time step through a learned weight matrix. This score is passed through a softmax function to normalize the importance scores into attention weights, which are then used to scale the LSTM outputs. This process allows the model to "attend" to the most relevant portions of the sequence, enhancing its ability to classify sounds with complex temporal patterns.

$$Attention(\mathbf{X}, \mathbf{W}, \mathbf{b}) = softmax(\tanh(\mathbf{XW} + \mathbf{b})) \quad (2)$$

Where $\mathbf{X}$ is the output from the LSTM layer and has shape [batch size, timestep, LSTM units] , $\mathbf{W}$ is the weight matrix of shape [LSTM units,1] and $\mathbf{b}$ is [timestep, 1].

After applying the attention mechanism, the output is fed into a second bidirectional LSTM layer, again with 128 units, which processes the temporally weighted features to further refine the temporal dependencies. This combination of LSTM and attention helps the model focus on the most significant temporal features and filter out irrelevant or noisy parts of the sequence.

The dense layers following the LSTMs are similar to the previous model, with a fully connected layer of 100 units before the final output layer, which contains 50 units corresponding to the classes in the ESC-50 dataset. The output is computed using a softmax activation function, which generates class probabilities.

VI. Results

In this section, we present the results obtained from evaluating the three models: CNN, CNN-LSTM, and CNN-LSTM-Attention, using three different approaches: original mel spectrograms, audio augmentation with mel spectrograms, and image augmentation.

A. Classification Task

The evaluation was performed using cross-validation on the ESC-50 dataset, and the performance of each model was measured in terms of the best classification accuracy. The results are shown in tab. 1.

The baseline CNN model, which only uses convolutional layers for feature extraction, achieved an accuracy of 57% when trained on the original mel spectrograms. With the addition of audio augmentation, the model saw a significant improvement, reaching an accuracy of 75%. However, when image augmentation techniques were applied, the model's performance decreased, achieving an accuracy of only 45%.

The CNN model has a total of approximately 2 million parameters, reflecting the complexity of the architecture.

The CNN-LSTM model, which introduces sequential modeling through LSTM layers, slightly outperformed the CNN model on the original dataset, achieving an accuracy of 58%. However, when audio augmentation was applied, this model performed slightly worse than the CNN model, reaching 69% accuracy. Interestingly, the CNN-LSTM model showed better performance than the CNN model when using image augmentation, with an accuracy of 54%. The total number of parameters for this model was approximately 1.38 million, which is lower than the CNN model due to the different architecture.

Finally, the CNN-LSTM-Attention model, which incorporates an attention mechanism to focus on the most relevant parts of the sequence, achieved the highest accuracy of 61% on the original dataset. When audio augmentation was applied, the model achieved 71% accuracy, showing that the attention mechanism helps in leveraging the augmented data. Similar to the CNN model, the performance decreased when using image augmentation, with an accuracy of 47%. The model has a comparable number of parameters to the CNN-LSTM model, approximately 1.38 million.

Overall, the results show that the CNN-LSTM-Attention model performed the best on the original data, while the CNN model excelled with audio augmentation. These results are also shown in the picture 7.

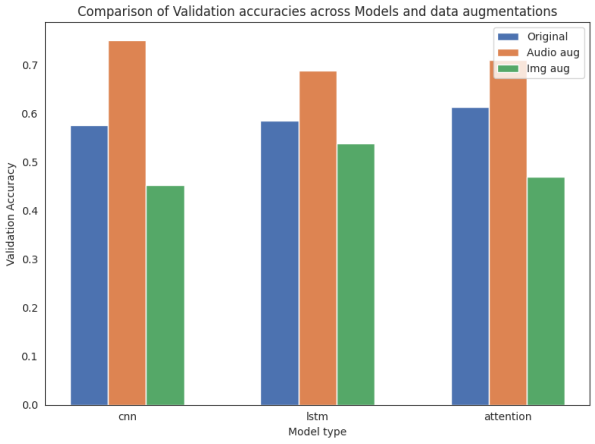


Fig. 7: comparison of models and different data augmentations techniques

The figure 8 illustrates the training and validation accuracy curves for the best-performing model, the CNN architecture, across 5 different cross-validation folds. Each fold is represented by two distinct curves, one for training accuracy and one for validation accuracy, depicting the model's learning behavior throughout the epochs.

As training progresses, the training curves show a consistent upward trend, with accuracy steadily increasing as the model optimizes its parameters for each fold. However, due to the implementation of early stopping, some training curves terminate earlier than others. This occurs when the validation accuracy fails to improve after a 15 epochs, indicating that the model has likely reached its peak performance on that fold, thus halting the training process to prevent overfitting.

	original	audio	img	params
cnn	0.57	**0.75**	0.45	2 M
lstm	0.58	**0.69**	0.54	1.4 M
attention	0.61	**0.71**	0.47	1.4 M

TABLE 1: Validation accuracies for different models for each preprocessing. The highest one for every model is obtained using audio augmentations on the dataset. LSTM and Attention model have similar number of parameters because they differ by only 263 parameters (the Attention parameters **W** and **b** in eq. 2).

The validation curves exhibit more fluctuation compared to the training curves, reflecting the model's ability to generalize to unseen data. In some folds, the validation accuracy steadily improves before reaching a plateau or slightly declining due to overfitting, triggering early stopping. Other folds show more variation in validation performance, highlighting the impact of different data partitions on the model's behavior.

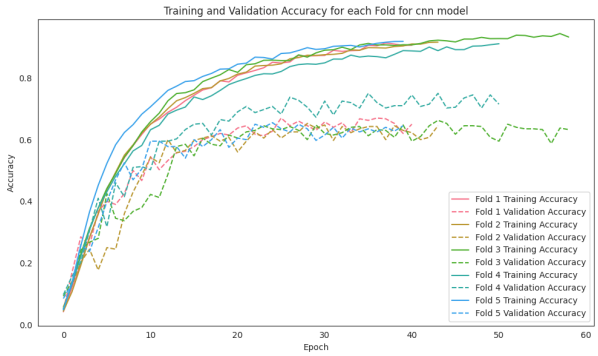


Fig. 8: Training curves for the best model: CNN trained with audio augmentated dataset.

B. Effect of Augmentations

In figure 9 we made a "deeper" analysis on the effects of choosing an augmentation technique.

Image augmentation consistently resulted in a decline in performance across all models, suggesting that image transformations are not as effective as the original dataset for this particular task. On the other hand, audio augmentations showed a consistent improvement in accuracy across all models. By applying transformations such as time-stretching, pitch-shifting, and adding noise, the models were able to generalize better and achieve higher accuracy. This improvement was observed in each of the models, demonstrating that augmenting the audio data directly helps the models capture more robust features, making audio augmentation a crucial step in this learning pipeline.

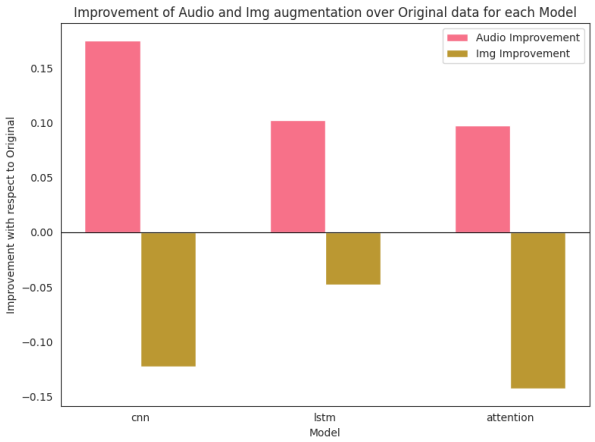


Fig. 9: Audio and Image improvement in accuracy over the original dataset for each model. In all cases the Audio augmentations helped the model to improve its performance and the exactly opposite can be said for the image augmentations.

C. Confusion Matrix

The confusion matrix in the Appendix reveals that the classes with the lowest prediction accuracy are ”snoring”, ”dog”, ”thunderstorm” and ”clapping” while the classes with the highest accuracy are ”insects”, ”rain” and ”crow.”

The poorer performance on ”snoring”, ”dog”, ”thunderstorm” and ”clapping” may be attributed to several factors. These classes often involve complex or overlapping acoustic features that can be challenging for models to distinguish clearly. For example, the sound of ”snoring” and ”dog” may have similar low-frequency components, leading to confusion in classification. Similarly, ”thunderstorm” encompasses a range of overlapping sounds (e.g., rain, thunder, wind) which can be difficult to disentangle, while ”clapping” may be easily confused with other abrupt or percussive sounds.

Conversely, ”insects,” ”rain,” and ”crow” are better predicted, likely due to their more distinct and less overlapping acoustic characteristics. The sound of ”insects” often has a consistent high-frequency pattern, ”rain” is characterized by a steady, consistent noise, and ”crow” has a relatively unique and recognizable pattern that stands out from other environmental sounds. These distinct features make it easier for the model to differentiate between these classes, resulting in higher accuracy.

VII. Concluding Remarks

In this paper we have implemented and compared three different neural network architectures: CNN, CNN-LSTM, and CNN-LSTM-Attention for ESC-50 classification. The experiments included three different data preparation techniques: standard mel spectrograms, audio augmentations, and image augmentations. The results showed that audio augmentations consistently improved model performance across all architectures, while image augmentations was exactly the opposite. This shows the importance of choosing the right augmentation techniques when addressing classification tasks. Also we showed that the addition of attention mechanism with a minimal increase of only 200 parameters led to a significant 2% improvement in accuracy.

However, there are still areas that can be improved upon. Implementing more sophisticated attention models, such as multi-head attention [13] or transformer-based architectures [14], could further boost the performance of these models as done by SOTA architectures. Additionally, the importance of audio augmentations highlights the necessity of high-quality and well-augmented datasets for training robust sound classification models. Future research could also focus on exploring better augmentation strategies and more diverse datasets to improve the generalizability and accuracy of environmental sound classification systems.

One of the biggest challenges in this research was working with a limited dataset. This constraint posed difficulties in training deep learning models, as such models typically require large amounts of data to achieve high performance. To mitigate overfitting, we employed cross-validation and data augmentation techniques. Another challenge was to implement an attention layer which was simple and intuitive to include in the model. This required careful considerations especially given the temporal nature of sound data.

References

[1] M. Vacher, J.-F. Serignat, and S. Chaillol, “Sound classification in a smart room environment: an approach using gmm and hmm methods,” 05 2007.

[2] F. Walden, S. Dasgupta, M. Rahman, and M. Islam, “Improving the environmental perception of autonomous vehicles using deep learning-based audio classification,” 2022.

[3] K. J. Piczak, “ESC: Dataset for Environmental Sound Classification,” in *Proceedings of the 23rd Annual ACM Conference on Multimedia*, pp. 1015–1018, ACM Press.

[4] K. Piczak, “Environmental sound classification with convolutional neural networks,” pp. 1–6, 09 2015.

[5] H. B. Sailor, D. M. Agrawal, and H. A. Patil, “Unsupervised filterbank learning using convolutional restricted boltzmann machine for environmental sound classification,” in *Interspeech 2017*, pp. 3107–3111, 2017.

[6] A. Guzhov, F. Raue, J. Hees, and A. Dengel, “Audioclip: Extending clip to image, text and audio,” 2021.

[7] B. Elizalde, S. Deshmukh, and H. Wang, “Natural language supervision for general-purpose audio representations,” 2024.

[8] S. Chen, Y. Wu, C. Wang, S. Liu, D. Tompkins, Z. Chen, and F. Wei, “Beats: Audio pre-training with acoustic tokenizers,” 2022.

[9] K. Chen, X. Du, B. Zhu, Z. Ma, T. Berg-Kirkpatrick, and S. Dubnov, “Hts-at: A hierarchical token-semantic audio transformer for sound classification and detection,” 2022.

[10] F. Font, G. Roma, and X. Serra, “Freesound technical demo,” in *Proceedings of the 21st ACM International Conference on Multimedia*, MM ’13, (New York, NY, USA), p. 411â€“412, Association for Computing Machinery, 2013.

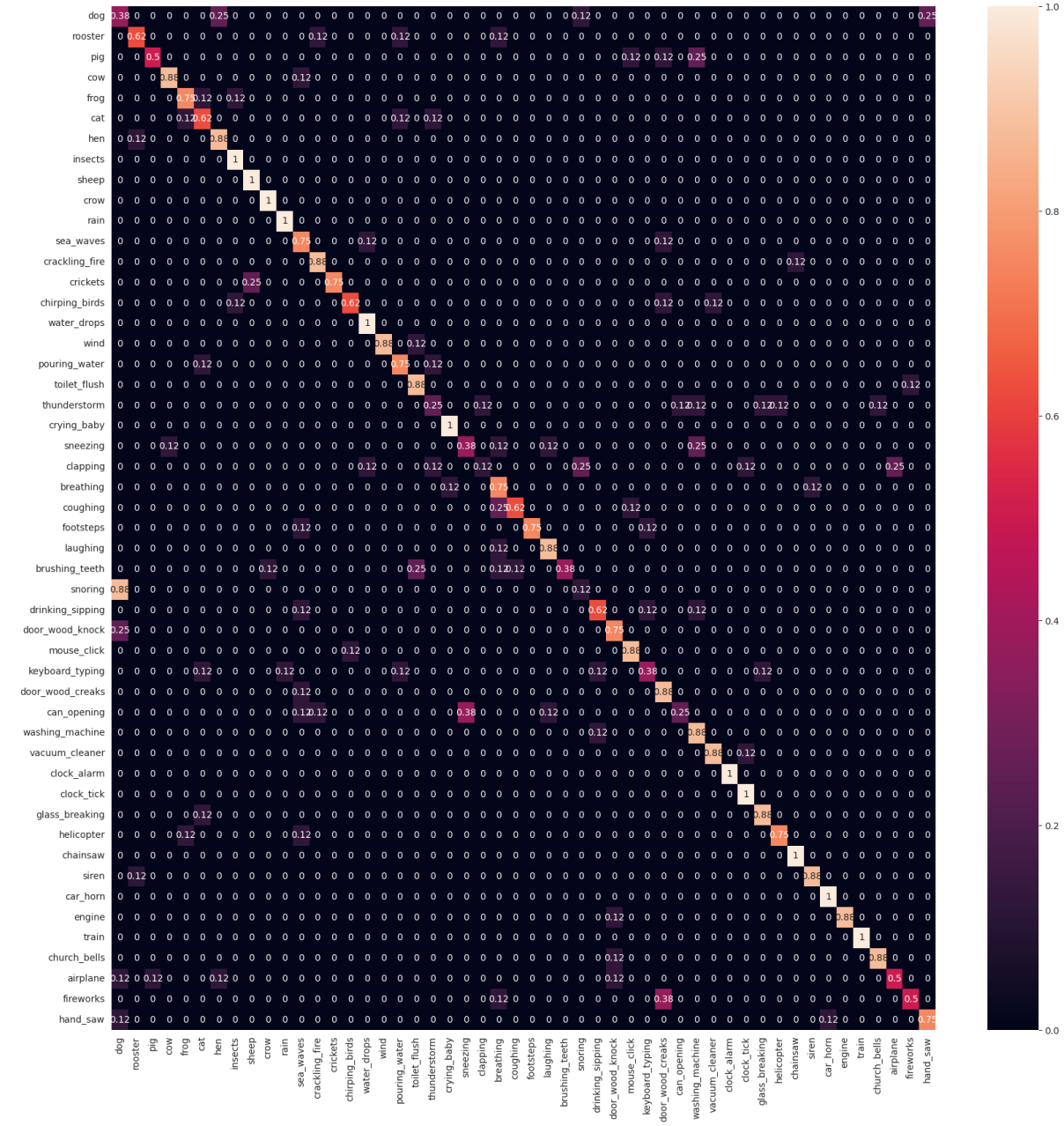
[11] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2017.

[12] C. Raffel and D. P. W. Ellis, “Feed-forward networks with attention can solve some long-term memory problems,” 2016.

[13] J.-B. Cordonnier, A. Loukas, and M. Jaggi, “Multi-head attention: Collaborate instead of concatenate,” 2021.

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” 2023.

APPENDIX


dog
rooster
pig
cow
frog
cat
hen
insects
sheep
crow
rain
sea_waves
crackling_fire
crickets
chirping_birds
water_drops
wind
pouring_water
toilet_flush
thunderstorm
crying_baby
sneezing
clapping
breathing
coughing
footsteps
laughing
brushing_teeth
snoring
drinking_sipping
door_wood_knock
mouse_click
keyboard_typing
door_wood_creaks
can_opening
washing_machine
vacuum_cleaner
clock_alarm
clock_tick
glass_breaking
helicopter
chainsaw
siren
car_horn
engine
train
church_bells
airplane
fireworks
hand_saw
1.0
0.8
0.6
0.4
0.2
0.0