Exploration of Machine Learning Techniques for Urban Sound Classification

Lorenzo Gentile[1]

[1]High Performance Computing Engineering, Politecnico di Milano

Abstract

Urban sound classification is a critical task in the field of sound classification, which involves identifying and categorizing various sounds in urban environments. In recent years, neural networks, particularly Convolutional Neural Networks (CNNs), have emerged as one of the most effective approaches for this task. In this study, we explore the performance of CNNs on the ESC-10 and UrbanSound8K datasets. We employ various techniques such as data augmentation, different audio features, and diverse model architectures. Our experiments utilize k-fold cross-validation to ensure robust evaluation. In several cases, our models achieve accuracy levels comparable to the state of the art.

Methodologies

Datasets

We used two datasets for our experiments:

1. **ESC-10**: This dataset is a subset of the ESC-50 dataset. It contains 400 samples divided into 10 classes. The sampling rate and audio duration are uniform across the dataset, with each clip being 5 seconds long and sampled at 44.1 kHz. The proposed models were also trained on the whole ESC-50 dataset to evaluate their performance on a larger and more diverse dataset.
2. **UrbanSound8K**: This is a larger dataset containing 8732 samples distributed across 10 classes. Unlike ESC-10, the audio duration and sampling rates in UrbanSound8K are not uniform (Figure 1). To standardize the samples before feeding them into a neural network, each clip is resampled to a common sampling rate (16KHz), and then either clipped or padded with zeros to ensure uniform length. Additionally, UrbanSound8K is not class-balanced, meaning that the number of samples varies across different classes. This imbalance can be mitigated during training by using class weights.

Data Augmentation

We employed a variety of data augmentation techniques to enhance the robustness and performance of our models. These techniques include:

1. **Noise Addition**: Various types of noise, such as white noise and pink noise, were added to the audio samples to increase diversity.
2. **Time Shifting**: This technique involves shifting the audio in time to make the model robust to temporal translations.
3. **Time Stretching**: This involves stretching or compressing the audio in time without altering its pitch.
4. **Pitch Shifting**: This technique shifts the pitch of the audio samples to create variations.

Each of these augmentation techniques can be activated or deactivated individually, and the extent of dataset growth is also a modifiable parameter. We observed that data augmentation was significantly more effective for the ESC-10 dataset, which is relatively small compared to UrbanSound8K. For the UrbanSound8K dataset, data augmentation did not result in any accuracy improvement and only slowed down the training process.

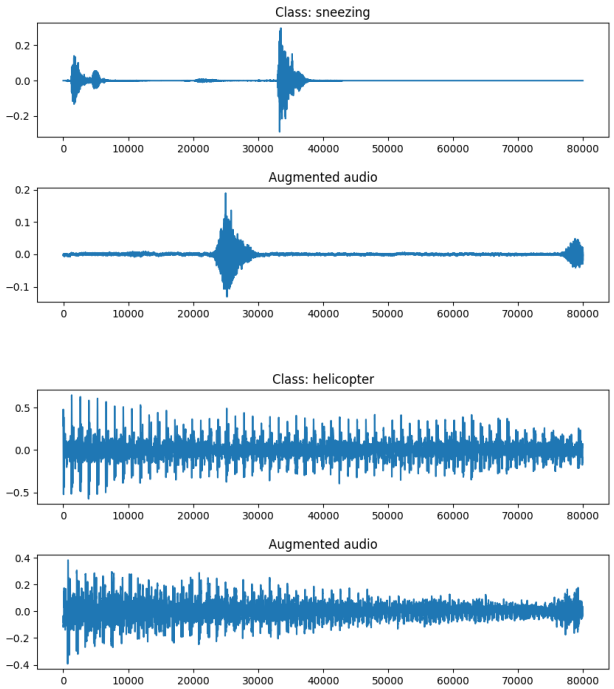
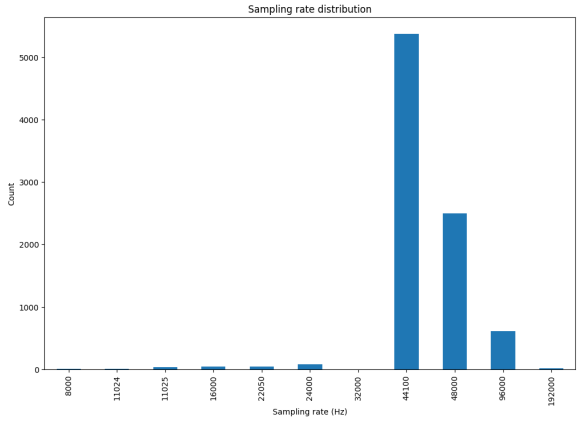
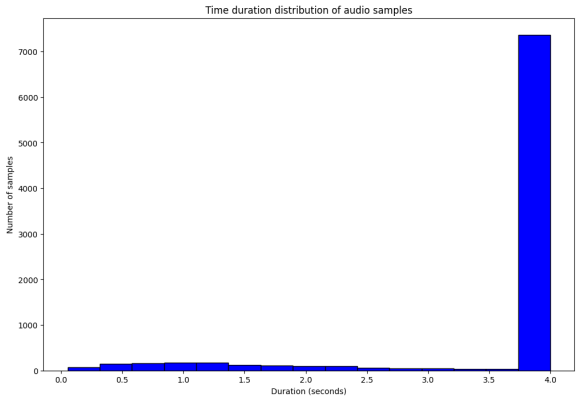


Figure 2: *Examples of data augmentation techniques applied to two audio samples from the ESC-10 dataset.*



(a) *Distribution of audio sample rates in the UrbanSound8K dataset.*



(b) *Distribution of time durations in the UrbanSound8K dataset.*

Figure 1: *Bar plots showing the non-uniformity in audio sample rates and time durations in the UrbanSound8K dataset.*

Audio Features

For the audio feature selection, we experimented with a wide range of audio transforms and features found in the literature. These are categorized into 1D and 2D features:

1D Features

- **Zero Crossing Rate (ZCR)**: Measures the rate at which the signal changes sign, indicating the noisiness of the signal.
- **MFCC 1D**: A flattened version of Mel Frequency Cepstral Coefficients, capturing the power spectrum of a sound.
- **Discrete Wavelet Transform (DWT)**: Decomposes the signal into different frequency components, useful for analyzing non-stationary signals.

2D Features

- **Classic Spectrogram (STFT)**: Uses Short-Time Fourier Transform to represent the signal in the time-frequency domain.
- **Mel Spectrogram**: A spectrogram where the frequencies are converted to the Mel scale, which is more aligned with human hearing.
- **Log-Mel Spectrogram**: Similar to the Mel Spectrogram but with decibel amplitudes, providing a logarithmic representation of the signal's power.
- **Mel Frequency Cepstral Coefficients (MFCC)**: Represents the short-term power spectrum of a sound, commonly used in speech and audio processing.
- **Scalogram**: Uses wavelet transform to represent the signal, capturing both frequency and temporal information.

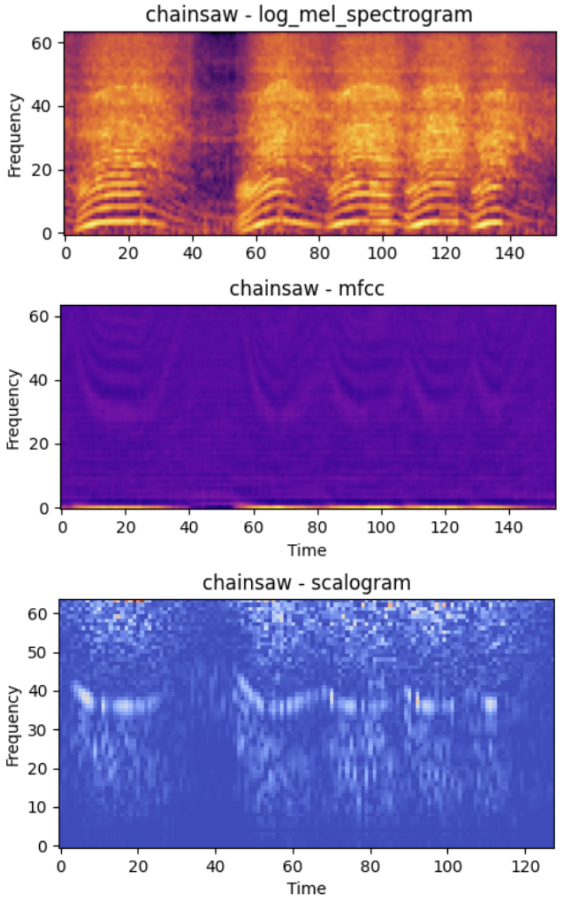


Figure 3: *Log-Mel Spectrogram, MFCC, and Scalogram of a chainsaw sound.*

Among these, the Mel Spectrogram and its logarithmic version (Log-Mel Spectrogram) generally provided the best results. These features are particularly effective when processed by Convolutional Neural Networks (CNNs), which excel at extracting 2D

patterns in the time-frequency domain. We also experimented with combining different features, which will be discussed further in the model architecture section.

Model Architecture

We chose a convolutional architecture for our model because it is widely considered one of the best options in the literature [1, 2]. The baseline architecture includes:

1. **Convolutional Layers:** Three convolutional layers, each followed by batch normalization and MaxPooling layers.
2. **Flattening:** The output from the convolutional layers is flattened.
3. **Dense Layers:** Two dense layers, each followed by batch normalization and dropout.
4. **Activation Functions:** ReLU activation is used in both the convolutional and dense layers.
5. **Output Layer:** A softmax layer, which outputs probabilities that are interpreted as the model's predictions (the highest probability corresponds to the predicted class).

This is the baseline setup, but all parameters can be changed in the notebook to see how the model responds.

We also tried using multiple inputs for the model, inspired by the work of [3], which proposed the AVCNN model. In our notebook, you can select both a 2D feature (like a mel spectrogram) and a 1D feature (like a discrete wavelet transform). The model combines the flattened output from the convolutional layers with the 1D input, and this combined vector is passed through the dense layers. This allows the model to use both visual and one-dimensional audio features in the learning process.

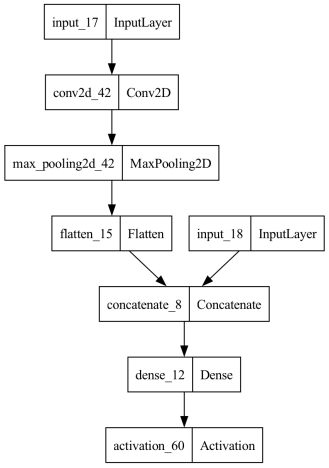


Figure 4: *2D and 1D feature combination in a simplified AVCNN like model.*

Additionally, you can select multiple 2D and 1D transforms. The model processes all the 2D inputs with separate convolutional branches and combines all the convolutional outputs with all the 1D inputs. This setup allows for many different configurations, but it should be noted that the model becomes more complex as more convolutional branches are added. With this setup, the task of finding a good architecture shifts to finding expressive audio features. For example, while log-mel spectrograms are often used, one might look for a 1D feature that complements the spectrogram (i.e., it captures aspects of the audio signal not well represented by spectrograms), potentially increasing accuracy compared to a spectrogram-only model.

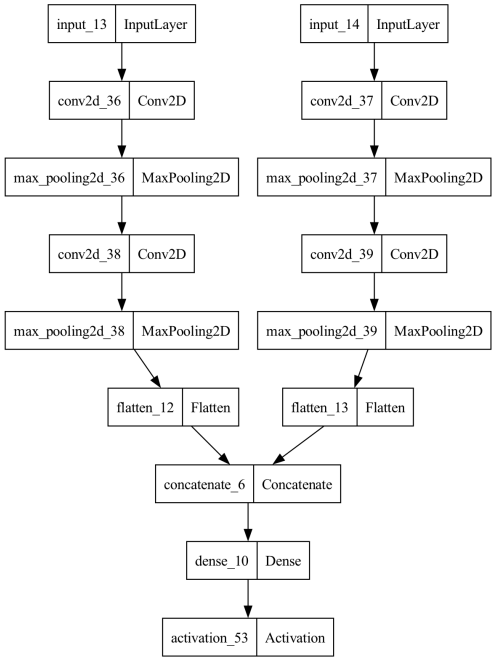


Figure 5: *2 convolutional branches for a double 2D input model.*

Results

Model Evaluation

For evaluating our models, we used 10-fold cross-validation on both datasets, as recommended by the dataset providers. Overall, our results were good and comparable to the state-of-the-art CNN models. However, we did not achieve the performance of more advanced techniques, such as those introduced in [4, 5], or the current state-of-the-art on urban sound datasets: the Audio Spectrogram Mixer with Roll-Time and Hermit FFT (ASM-RH), introduced in [6].

In Table 1, we present the 10-fold accuracies for our best models. Our highest accuracy was 75% on

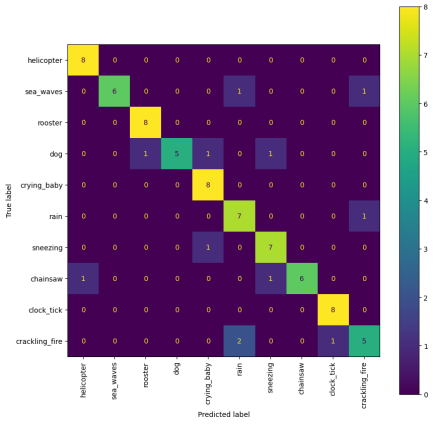
Dataset	Representation	Data Augmentation	K-fold Accuracy
UrbanSound8K	log-mel-spectrogram	No	74.14%
UrbanSound8K	scalogram	No	62.01%
UrbanSound8K	log-mel-spectrogram + scalogram	No	75.27%
ESC-10	log-mel-spectrogram	Yes	85.00
ESC-50 6	log-mel-spectrogram	Yes	54.06%

Table 1: *List of best models and their k-fold accuracies.*

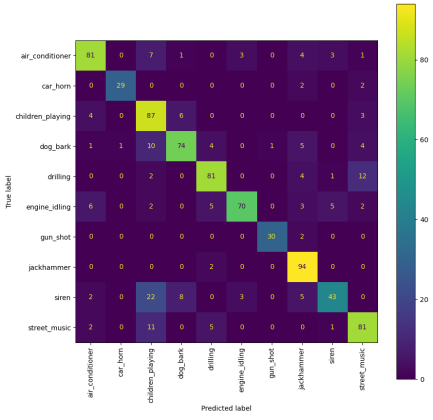
UrbanSound8K, which is slightly lower than the 79% achieved by SB-CNN [2], the best *pure* CNN model we found in the literature. For ESC-10, our best 5-fold accuracy was 85%.

Model Inspection

From the confusion matrices of the best models (see Figure 6), we observe that in the ES-10 dataset (a very small dataset), the most confused pair is rain and crackling fire. For the UrbanSound8K dataset, the most confused pairs are siren and children playing, street music and children playing, and drilling and street music. Notably, gunshot is almost never confused with any other sound.



(a) *Confusion matrix for ESC-10.*



(b) *Confusion matrix for UrbanSound8K.*

Figure 6: *Confusion matrices for the best models on the ESC-10 and UrbanSound8K datasets.*

We also examined the internal workings of the models. In the inference notebook, you can load a model from the `saved_models` folder and plot the activations of the hidden layers of the neural network. This allows us to see the patterns that each layer has learned to recognize and understand how data flows through the model (see Figures 7 and 8). Additionally, you can print the filters of the convolutional layers, which can provide insights into what these layers are detecting in an image.

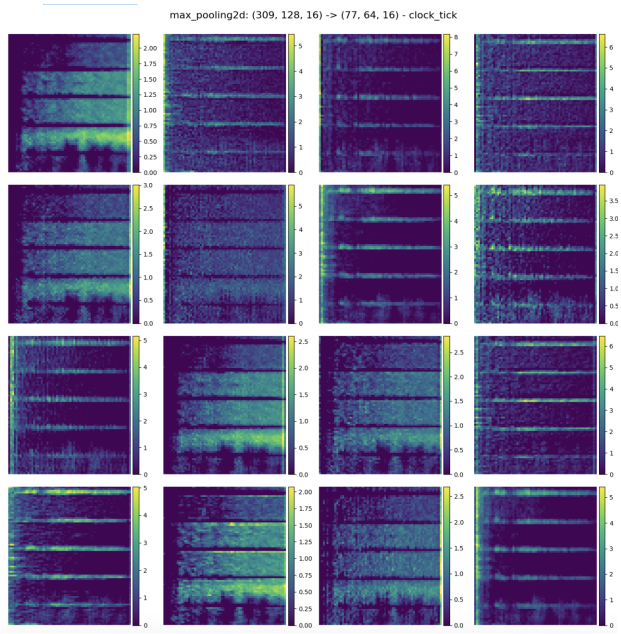


Figure 7: *Output of one of the hidden convolutional layers of the neural network.*

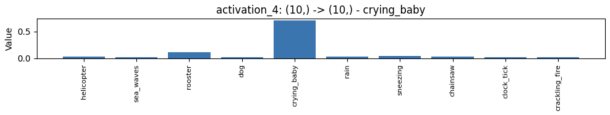


Figure 8: *Output of the final layer of the neural network. (Softmax layer). The model is correctly predicting the class of the input.*

References

[1] Massoud Massoudi, Siddhant Verma, and Riddhima Jain. "Urban Sound Classification using CNN". In: *2021 6th International Conference on Inventive Computation Technologies (ICICT)*. 2021, pp. 583–589. DOI: 10.1109/ICICT50816.2021.9358621.

[2] Justin Salamon and Juan Pablo Bello. "Deep Convolutional Neural Networks and Data Augmentation for Environmental Sound Classification". In: *IEEE Signal Processing Letters* 24.3 (2017), pp. 279–283. ISSN: 1558-2361. DOI: 10.1109/lsp.2017.2657381. URL: http://dx.doi.org/10.1109/LSP.2017.2657381.

[3] Turgut Özseven. "Investigation of the effectiveness of time-frequency domain images and acoustic features in urban sound classification". In: *Applied Acoustics* 211 (2023), p. 109564. ISSN: 0003-682X. DOI: 10.1016/j.apacoust.2023.109564. URL: https://www.sciencedirect.com/science/article/pii/S0003682X23003626.

[4] Avi Gazneli et al. *End-to-End Audio Strikes Back: Boosting Augmentations Towards An Efficient Audio Classification Network*. 2022. arXiv: 2204.11479 [cs.SD].

[5] Andrey Guzhov et al. *ESResNet: Environmental Sound Classification Based on Visual Domain Models*. 2020. arXiv: 2004.07301 [cs.CV].

[6] Qingfeng Ji, Yuxin Wang, and Letong Sun. *Mixer is more than just a model*. 2024. arXiv: 2402.18007 [cs.LG].