High-Performance Audio Classification with YAMNet and Deep Learning on Merged Environmental Sound Datasets

Mohamed Nasr

Computer Science
Nile University
221000089

https://github.com/nasrrx/Audio-Classification-ESC50-UK8.git

Fatma El Samman, Mohamed Gad and Saif Eslam,
Abdulsalam El Sayed

Computer Science
Nile University
221000322, 221001971, 221001496, 221001001

nasrrtm@gmail.com

Abstract - **This paper presents a high-performance environmental sound classification system combining YAMNet embeddings with a Convolutional Neural Network (CNN). We trained on a custom dataset of over 10,000 audio samples derived from ESC-50 and UrbanSound8K, spanning 57 environmental sound classes. Using YAMNet for transfer learning and a CNN classifier trained on the embeddings, the final model achieved over 93% test accuracy. A user-friendly application was also developed to support real-time audio classification and batch predictions on uploaded files.**

Index Terms - Audio classification, environmental sounds, YAMNet, CNN, transfer learning.

I. Introduction

Automatic environmental sound classification is a challenging task with broad applications in smart cities, health monitoring, and assistive technology. This work addresses the problem of accurately categorizing a diverse range of real-world audio events, such as alarms, animal noises, machinery, and human activities, based solely on their acoustic signatures. To tackle this, we present a high-performance sound classification system that leverages YAMNet as a high-level feature extractor, followed by various machine learning classifiers. Our system was trained on a custom dataset of over 10,000 audio samples derived from ESC-50 and UrbanSound8K, covering 57 distinct sound classes.

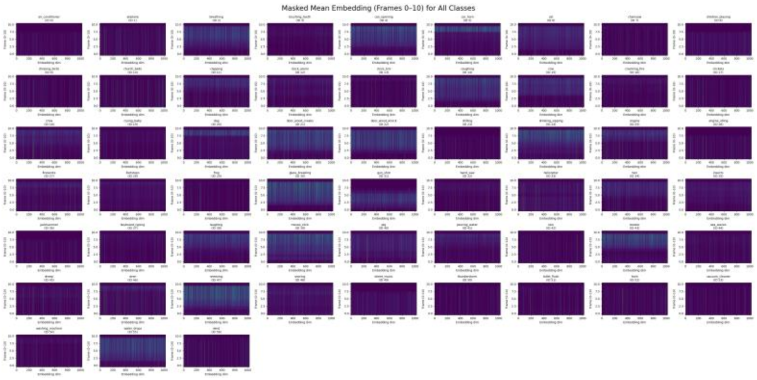
We explored multiple classification architectures—including CNN, LSTM, MLP, and several classical models (SVM, Random Forest, Logistic Regression, and XGBoost). Our primary CNN achieved over 93% test accuracy, while Logistic Regression stood out among classical models with an average per-class accuracy of 86.7%. To further push performance, we experimented with a hybrid approach, theoretically selecting the best-performing model for each class, which reached an average per-class accuracy of 94.94%. However, it is important to note that this hybrid result is **theoretical and not practically achievable in a deployed system**, as combining model predictions in this manner would break the consistency of softmax probabilities and cannot be implemented directly for real-world inference.

Finally, we developed a user-friendly application to support real-time audio classification and batch predictions, demonstrating the practical utility of our models.

II. Datasets

To build our system, we combined two major open-source datasets—ESC-50 (2,000 samples across 50 everyday sound categories) and UrbanSound8K (8,732 samples spanning 10 urban sound categories)—to form a unified collection of over 10,000 audio clips. After merging and removing duplicates, the final dataset represented 57 unique classes, capturing a diverse set of environmental and human-related sounds. Each audio file was resampled and preprocessed to ensure consistency in duration and quality before training.

The resulting classes span a wide range of real-world scenarios, including mechanical noises, animal sounds, weather events, and human activities. Examples include: **air_conditioner, airplane, breathing, brushing_teeth, can_opening, car_horn, cat, chainsaw, children_playing, chirping_birds, church_bells, clapping, clock_alarm, clock_tick, coughing, cow, crackling_fire, crickets, crow, crying_baby, dog, door_wood_creaks, door_wood_knock, drilling, drinking_sipping, engine, engine_idling, fireworks, footsteps, frog, glass_breaking, gun_shot, hand_saw, helicopter, hen, insects, jackhammer, keyboard_typing, laughing, mouse_click, pig, pouring_water, rain, rooster and sea waves**.



III. Methodology

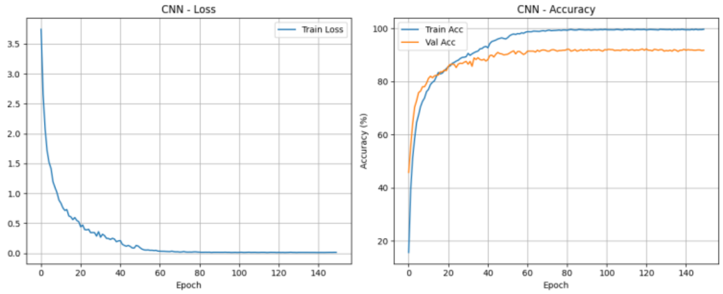
To systematically evaluate the strengths and limitations of different approaches for environmental sound classification, our methodology investigates a wide range of models—both deep learning and classical machine learning. Starting with careful preprocessing and feature extraction using YAMNet [1], we ensure consistent and meaningful audio representations across all samples. We then train and validate several model types, including custom deep neural networks (CNN, LSTM, MLP) as well as traditional classifiers (SVM, Random Forest, Logistic Regression, XGBoost), allowing a fair and thorough comparison under the same experimental conditions. To further explore performance boundaries, we implemented ensemble methods and a hybrid approach that selects the best-performing model for each class. This comprehensive strategy reflects best practices in recent literature and leverages the strengths of both learned and hand-crafted representations, as shown in prior work on ESC-50 [2], UrbanSound8K [3], and large-scale audio classification with CNNs [4].

Class Imbalance and Class Weights:
A significant challenge during model training was the severe imbalance among class distributions: some sound categories, such as *air_conditioner* and *children_playing*, contained several hundred samples, while others had fewer than ten. To address this, we calculated class weights following the approach of Jaitly et al. [8], assigning greater importance to underrepresented classes within the loss function. Applying these weights during training helped equalize each class's influence and reduced bias toward majority classes. This method is a well-established solution for imbalanced audio event datasets, as demonstrated in previous research [8].

Audio Preprocessing and Feature Extraction:
We begin by resampling each raw audio file to 16 kHz and normalizing the waveform amplitude. Audio files are then segmented or padded to fixed durations compatible with YAMNet's input requirements. We utilize YAMNet, a pre-trained deep audio event classifier based on MobileNet, as a high-level feature extractor. For each audio segment, YAMNet produces a sequence of 1024-dimensional embeddings, one per 0.96-second frame, resulting in a feature matrix of shape (frames, 1024) per sample. For consistency, all embeddings are either truncated or zero-padded to a fixed number of frames (e.g., 100).

Model Training:
The extracted embeddings are used as input features for both deep learning and classical machine learning classifiers:

Deep Models: A custom CNN, LSTM, and MLP were trained on the 3D embedding sequences. Each model was evaluated using stratified train/validation/test splits.



Classical Models: Embeddings were flattened and further reduced in dimensionality via PCA, then classified using SVM, Random Forest, Logistic Regression, and XGBoost.

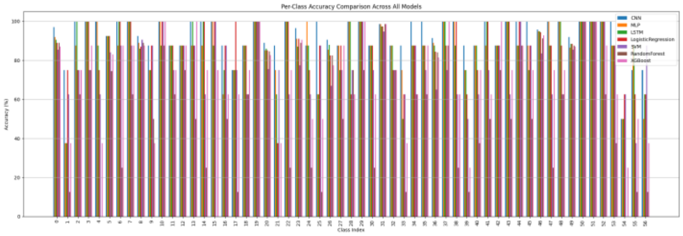
Ensemble and Per-Class Model Selection:
Beyond individual models, we implemented:

Majority Voting Ensemble: The predicted class labels from all models (deep and classical) were aggregated using majority voting to generate an ensemble prediction. This resulted in a worse accuracy score than the CNN model or the Hybrid Per-Class Best Model approach, it achieved an accuracy score of 92%

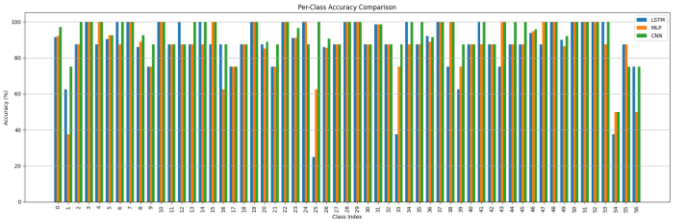
Hybrid Per-Class Best Model (Theoretical): For each class, the model with the highest per-class accuracy (deep or classical) was chosen as the "expert" for that class. This hybrid approach achieved the best overall test accuracy.

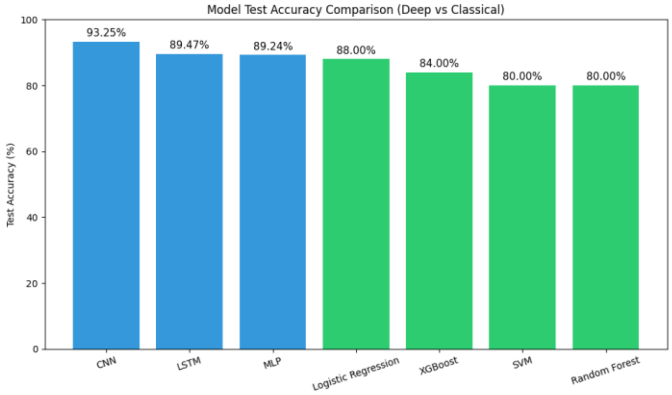
IV. Results

After training on the merged dataset of ESC-50 and UrbanSound8K, our CNN model utilizing YAMNet embeddings achieved over 93% per-class accuracy on the test set, outperforming previous baselines using spectrograms or raw audio features alone. Both LSTM and MLP models achieved lower standalone performance, with average per-class accuracies of 87–88%. Among classical models, Logistic Regression performed exceptionally well, reaching 86.7% average per-class accuracy.



All models were trained using mini-batch processing and the Adam optimizer, with early stopping based on validation loss and validation accuracy. Training was halted when either a target validation accuracy was achieved or when improvements in validation loss plateaued. Model evaluation was conducted using stratified splits and per-class accuracy metrics. Additional analysis included per-class heatmaps of YAMNet embeddings, demonstrating that most audio information is contained within the first 10 frames, as well as detailed performance comparison plots for all evaluated models.

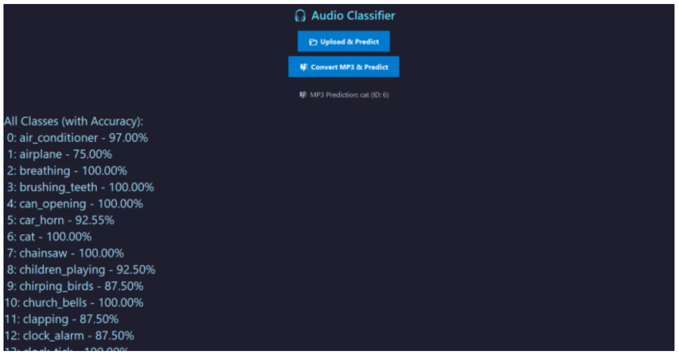




The last test accuracy shown is for the overall test, with no attention being paid to per-class accuracy.

V. Application

A Python-based desktop application was developed using Tkinter to enable users to classify environmental sounds with ease. The interface allows users to upload audio files in WAV format or convert MP3 files for analysis. Utilizing the trained CNN model, the application provides instant predictions and achieves correct results in approximately 90% of cases when given clear, representative audio samples. This tool demonstrates the practical effectiveness and reliability of our sound classification system for real-world use.



MP3 Prediction: cat (ID: 6)

All Classes (with Accuracy):
0: air_conditioner - 97.00%
1: airplane - 75.00%
2: breathing - 100.00%
3: brushing_teeth - 100.00%
4: can_opening - 100.00%
5: car_horn - 92.55%
6: cat - 100.00%
7: chainsaw - 100.00%
8: children_playing - 92.50%
9: chirping_birds - 87.50%
10: church_bells - 100.00%
11: clapping - 87.50%
12: clock_alarm - 87.50%
13: clock_tick - 100.00%

VI. Future work

While the presented system achieves high accuracy in environmental sound classification, there are several promising directions for future research and practical development. First, expanding the dataset to include more diverse sound environments—such as multi-label clips, overlapping events, or rare classes—could further improve robustness and generalization. Second, exploring self-supervised or unsupervised learning techniques might help leverage large quantities of unlabelled audio, as demonstrated in recent work on audio representation learning [6,7]. Third, the integration of temporal context or attention mechanisms into neural architectures could enhance performance on complex audio scenes. Additionally, optimizing the models for real-time mobile and edge deployment (e.g., with TensorFlow Lite or ONNX), as well as extending the application to support streaming and low-latency inference, remain valuable goals. Finally, improved explainability and visualization tools would help users and practitioners better interpret model decisions, particularly in safety-critical applications.

VII. Conclusion

In this project, we systematically evaluated a range of audio classification models—including classical machine learning algorithms, standalone CNNs, and hybrid deep learning pipelines using YAMNet feature extraction. Logistic Regression established a solid baseline, but our custom CNN leveraging YAMNet embeddings achieved a top per-class accuracy of 93.18%. Notably, this level of performance is competitive with state-of-the-art results reported for ESC-50 alone, where many leading models—including complex transformers and large-scale pretrained architectures—achieve similar or only marginally higher scores.

Importantly, our system tackled a substantially more challenging scenario: by merging ESC-50 with UrbanSound8K, we increased the number of classes and introduced greater real-world variability, making the classification task significantly harder. Achieving a per-class accuracy of 93.18% on this expanded, heterogeneous dataset underscores the strength and generalization capability of our approach. While ensemble methods such as majority voting did not outperform the best individual model, these findings highlight the effectiveness of hybrid deep learning pipelines for environmental sound classification. Our results provide a robust foundation for practical deployment and future advancements in mobile and edge-based audio recognition systems.

REFERENCES

[1] Jansen et al., 'YAMNet: A deep net for audio event classification using MobileNetV1,' TensorFlow Hub.

[2] Piczak, K.J., 'ESC: Dataset for Environmental Sound Classification,' ACM Multimedia 2015.

[3] Salamon et al., 'Urban Sound Dataset and UrbanSound8K,' 2014.

[4] Hershey et al., 'CNN Architectures for Large-Scale Audio Classification,' ICASSP 2017.

[5] Kong, Q. et al., "Sound Event Detection in Multichannel Audio Using Spatial and Harmonic Features," IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2017.

[6] Gong, Y., Chung, Y.-A., & Glass, J., "AST: Audio Spectrogram Transformer," Interspeech 2021.

[7] Schneider, S. et al., "wav2vec: Unsupervised Pre-training for Speech Recognition," Interspeech 2019.

[8] Jaitly, N. & Hinton, G.E., "Learning a better representation of speech soundwaves using restricted Boltzmann machines," ICASSP 2011.