Audio Classification on ESC-50 Using CNN and Transformer Architectures

David Marzban
KTH Royal Institute of Technology
dmarzban@kth.se

Bashar Levin
KTH Royal Institute of Technology
basharl@kth.se

Johannes Joujo
KTH Royal Institute of Technology
joujo@kth.se

Nasir Alizade
KTH Royal Institute of Technology
alizade@kth.se

Abstract

This project explores environmental sound classification on the ESC-50 dataset using a hybrid architecture that combines convolutional neural networks (CNNs) with transformer-based models. We propose a model that integrates a CNN frontend with a pre-trained GPT-2 encoder, repurposed for audio tasks. To evaluate the impact of transfer learning, we compare this model against a baseline with the same architecture but using a randomly initialized GPT-2. The fine-tuned model achieved a test accuracy of 55% $\pm$ 2.1, significantly outperforming the baseline model, which reached only 35% $\pm$ 2.4. Both models performed poorly before training, with accuracies of 2.0%. These results demonstrate that incorporating pre-trained language models into audio pipelines and progressive fine-tuning and regularization techniques can substantially improve classification performance.

1 Introduction

1.1 Dataset

The ESC-50 [1] dataset was used. It consists of 5-second audio clips sampled at 16kHz, labeled across 50 environmental sound categories. These categories span a wide range of everyday sounds, grouped into five major groups: animals (e.g., dog barking, rooster), natural soundscapes (e.g., rain, wind), human non-speech sounds (e.g., sneezing, coughing), interior/domestic sounds (e.g., vacuum cleaner, clock alarm), and exterior/urban noises (e.g., siren, helicopter). This diversity makes ESC-50 a suitable benchmark for general-purpose environmental sound classification.

1.1.1 Preprocessing

The audio files from the ESC-50 dataset were first resampled to 16kHz and converted into log-Mel spectrograms using 40 mel bands. To enhance generalization, raw waveform augmentations including additive Gaussian noise, random pitch shifting, and time stretching was applied to the original data. Additionally, SpecAugment was used to mask random time and frequency bands within the spectrograms.

Each spectrogram was normalized per sample and padded or truncated to a fixed length of 100 frames. Two-channel inputs were constructed by stacking the original and augmented spectrograms. Finally, the data was split into training, validation, and test sets using stratified sampling, and serialized into PyTorch tensors for efficient loading during training. Figure 1 illustrates the preprocessing pipeline, showing a Log-Mel Spectrogram (Channel 1) and its augmented version (Channel 2), which highlight the spectral characteristics and augmentation effects.

30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

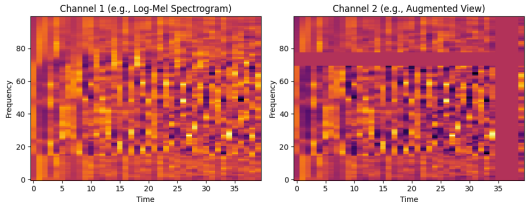

Figure 1: Log-Mel Spectrogram (left) and augmented view (right) of an ESC-50 audio sample, demonstrating the preprocessing pipeline.

2 Method

This section describes the model architecture, training strategy, and hardware setup used to perform classification on the ESC-50 dataset. Two models are considered: a fine-tuned model utilizing a pre-trained GPT-2 transformer and a baseline model with the same architecture but using a randomly initialized GPT-2. The baseline serves as a control for evaluating the effectiveness of transfer learning in audio classification.

2.1 Architecture

Both the baseline and the proposed model adopt a hybrid CNN-GPT-2 architecture [2]. The model begins with a convolutional frontend that processes 2-channel log-mel spectrograms. It consists of two convolutional blocks with batch normalization, ReLU activation, max pooling, and dropout. The CNN output is reshaped and projected using a linear layer to match the GPT-2 embedding dimension.

This token sequence is then passed through a GPT-2 transformer. The baseline model uses a randomly initialized GPT-2 model, while the fine-tuned model uses a pre-trained GPT-2 from HuggingFace (GPT-2). The final output is obtained by mean-pooling the transformer outputs and passing them through a two-layer MLP classifier [3].

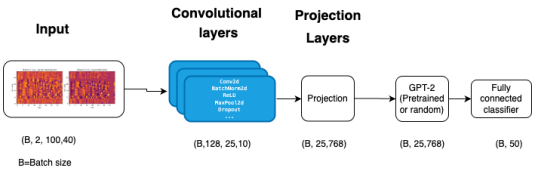

Figure 2: Tensor shape flow through the model architecture.

Figure 2 illustrates how each input audio sample is represented as a 2-channel log-mel spectrogram of shape (2, 100, 40). This is processed by a CNN, reshaped, projected into GPT-2 embedding space, encoded contextually by GPT-2, and finally classified through a feedforward network into one of 50 ESC-50 classes. Shapes shown are per batch: (B, ...).

2.2 Training

Both models were trained using AdamW optimizer with a weight decay of $1e-2$. A ReducedLROnPlateau learning rate scheduler was applied based on validation accuracy to reduce the learning rate when the performance plateaued. Training was conducted using a batch size of 32 for 200 epochs.

The fine-tuned model, the GPT-2 backbone, was initially partially trainable, with only the top 2 transformer blocks unfrozen at the beginning of training. Additional blocks were unfrozen at specific epochs [0,30,60,90,120,140], following a schedule [2,4,6,8,10,12] to increase the number of trainable layers. This allowed for gradually adapting the pre-trained GPT-2 model to the audio classification task.

The loss function used was KL-Divergence over log-softmax predictions. Label smoothing (0.05) was applied to improve generalization. Additionally, the fine-tuned model used Mixup augmentation [4] with alpha=0.4, blending input-label pairs to further regularize training.

The baseline model, by contrast, used a randomly initialized GPT-2 and did not employ progressive unfreezing or Mixup. It was trained with a smaller learning rate ($1e-5$) and full parameter optimization.

2.3 Hardware setup

Most training was conducted on a KTH-provided remote server equipped with an NVIDIA H100 80GB GPU (MIG 1g.20gb). This high-memory setup enabled efficient training of large models like GPT-2, particularly during the fine-tuning phases, without encountering memory bottlenecks.

3 Experiments

To evaluate the effectiveness of using a pre-trained language model for environmental sound classification, we compared two variants of the CNN-GPT2 architecture:

- Baseline model: A CNN followed by a randomly initialized GPT-2 with 6 transformer blocks, trained without augmentation.
- Fine-tuned model: Same architecture, but initialized with pre-trained GPT-2 weights and trained using progressive layer unfreezing, Mixup augmentation, and label smoothing.

3.1 Design Choices and Observations

Initial training runs using the fine-tuned GPT-2 model without regularization led to overfitting. The model quickly achieved near-perfect training accuracy, while validation accuracy plateaued early. To address this:

- Label smoothing was introduced to soften hard labels and reduce overconfidence.
- Mixup augmentation was added to improve generalization by interpolating both inputs and labels.

These strategies greatly improved validation performance and were adopted for all subsequent experiments involving the fine-tuned model.

We also experimented with different learning rate schedulers to find the most effective one for this setup. The following were tested:

- ReduceLROnPlateau (final choice): Reduced the learning rate when validation accuracy stopped improving, also improved the validation accuracy [5].
- CosineAnnealingLR: Performed reasonably well but was less stable during progressive unfreezing.
- OneCycleLR: This provided faster early learning but was sensitive to initial learning rate settings, resulting in the worst performance among the three options.

Among these, ReduceLROnPlateau gave the best balance of stability and convergence when combined with progressive unfreezing and Mixup. It was therefore used in all the final training runs.

3.2 Evaluation

Both models were trained for 200 epochs using a batch size of 32. Each configuration was evaluated using the same train/validation split. The model was saved based on the best validation accuracy. Training dynamics were logged using TensorBoard.

4 Results

We evaluated the test performance of both the baseline and fine-tuned CNN-GPT2 models across seven independent runs. Figure 3 summarizes the mean and standard deviation of test accuracies before and after training for both models.

The fine-tuned model, which incorporates a pre-trained GPT-2 encoder and Mixup augmentation and label smoothing, achieved a mean test accuracy of $55.9\% \pm 2.1\%$. In contrast, the baseline model, initialized randomly with a GPT-2 backbone, achieved only $36.4\% \pm 2.4\%$ accuracy. Before training, both models performed poorly, with approximately 2.0% test accuracy.

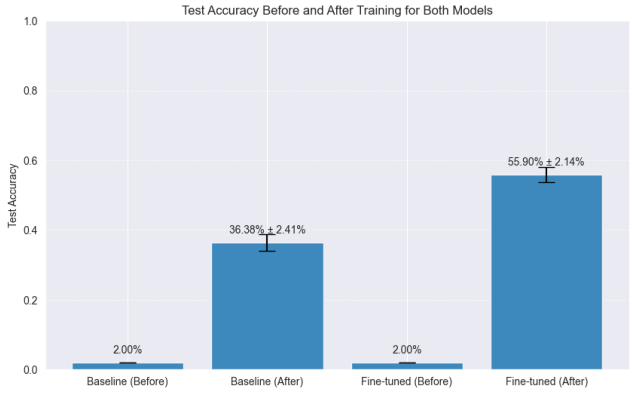

Figure 3: Test accuracy before and after training for baseline and fine-tuned models. Error bars indicate standard deviation over 7 runs.

Figure 4 shows the confusion matrices of both models on the ESC-50 test set. The fine-tuned model yields a clearer diagonal and fewer off-diagonal errors, demonstrating stronger discriminative ability. Notably, it improved predictions for overlapping categories such as “siren”, “dog”, and “clock alarm”, which the baseline model frequently misclassified.

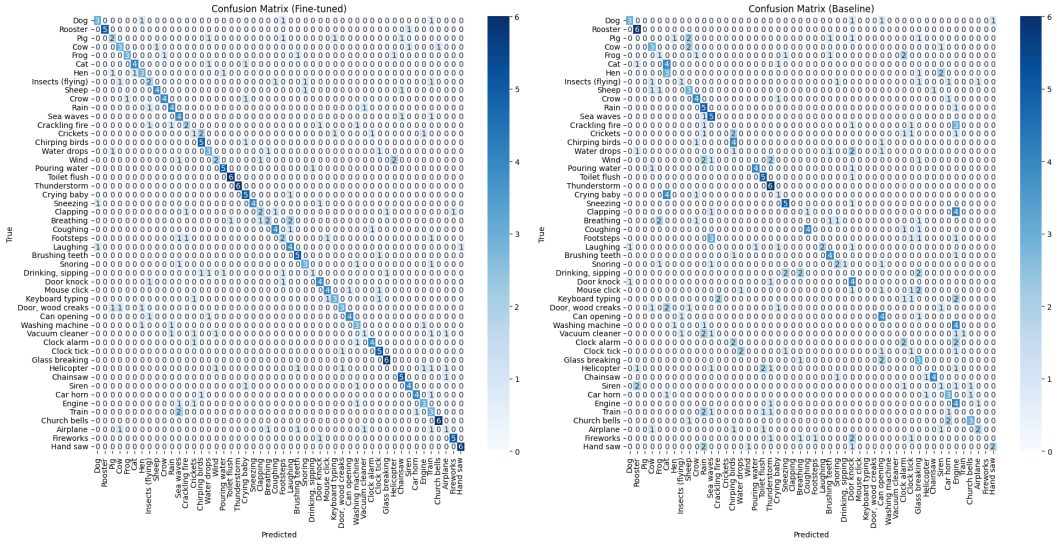

Figure 4: Confusion matrices for fine-tuned (left) and baseline (right) models. Each cell shows the number of samples predicted per class.

We also visualized the training and validation loss and accuracy over 200 epochs for both models (Figures 5 and 6). The fine-tuned model showed steady improvement and smoother convergence, with validation accuracy continuing to rise. In contrast, the baseline model plateaued early and underfitted the training data, with a large gap between training and validation accuracy.

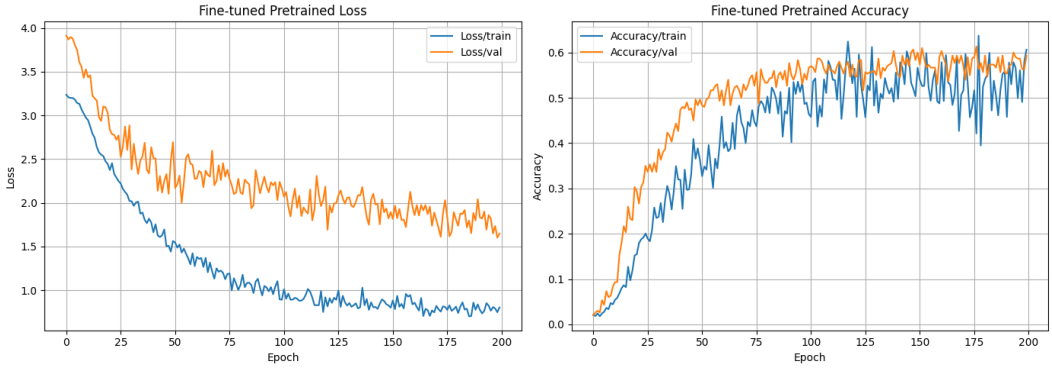

Figure 5: Training and validation loss (left) and accuracy (right) for the fine-tuned model. The use of regularization improved generalization.

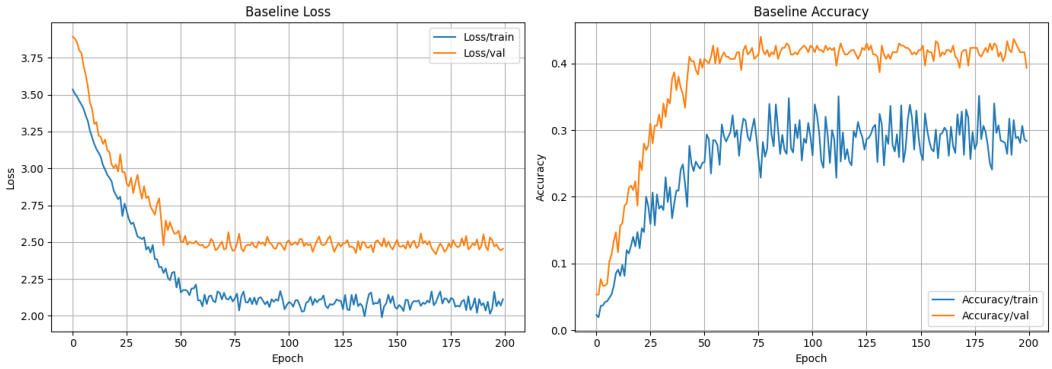

Figure 6: Training and validation loss (left) and accuracy (right) for the baseline model. The model quickly underfitted, with train accuracy saturating.

5 Discussion and Conclusions

The results from our experiments demonstrate that incorporating a pre-trained GPT-2 transformer into an audio classification pipeline can provide tangible benefits over using a randomly initialized transformer model. Specifically, the fine-tuned GPT-2 model achieved a best validation accuracy of 60.33(%), compared to 55.00(%) for the same architecture without regularization and transfer learning. This supports our hypothesis that transfer learning using large language models can improve classification performance in the audio domain.

In conclusion, this project demonstrates that a hybrid CNN-GPT-2 model, when properly regularized and fine-tuned, can outperform simpler baselines on environmental sound classification tasks. While the absolute improvement in validation accuracy may appear modest, it is significant given the challenges of generalizing from a relatively small and diverse audio dataset.

References

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

[2] S. Hershey, S. Chaudhuri, D. P. W. Ellis, *et al.*, "Cnn architectures for large-scale audio classification," 2017. Accessed: 2025-05-18.

[3] OpenAI Community, "Gpt2 - a large-scale generative language model," 2024. Accessed: 2025-05-18.

[4] Y. N. D. D. L.-P. Hongyi Zhang, Moustapha Cisse, "mixup: Beyond empirical risk minimization," 2018. Accessed: 2025-05-18.

[5] V. Chugani, "A gentle introduction to learning rate schedulers," 5 2025. Accessed: 2025-05-18.