Utilizing Transfer Learning for Environmental Sound Classification

Shubh Jaroria
Department of Computer Science
State University of New York at Buffalo
Buffalo, NY 14260
shubhjar@buffalo.edu

1 Team Information

This team comprises of only ONE person, namely, the author on the top, Shubh Jaroria. Please find the code and instructions to run it at https://github.com/ShubhJaroria/Transfer-Learning-YAMNet .

2 Introduction

Environmental sound classification is an important task with various applications, including surveillance, acoustic monitoring, and human-computer interaction. However, it is also a challenging task due to the complex nature of sound signals and the diversity of environmental sounds. To address this challenge, machine learning techniques have been widely used in recent years. In particular, transfer learning has shown great promise in improving the performance of sound classification models.

Transfer learning [4] is a machine learning technique in which a model is trained on a large dataset and then fine-tuned on a smaller, related dataset for a specific task. This technique is particularly useful in sound classification, where the availability of large labeled datasets is limited. In this project, we explore the use of transfer learning with YAMNet for environmental sound classification on the ESC-50 dataset.

YAMNet [2] is a deep learning model developed by Google for sound classification tasks. It uses a convolutional neural network (CNN) to extract features from sound signals and a classifier to predict the sound class. YAMNet has been pre-trained on a large-scale dataset called AudioSet, which contains millions of 10-second sound clips labeled with over 500 sound classes. AudioSet covers a wide range of sound categories, including musical instruments, animal sounds, and human sounds.

The ESC-50 dataset [3] is a publicly available dataset of environmental sounds, consisting of 2,000 audio recordings from 50 sound classes, i.e, 40 recordings for each class. Each recording is five seconds long and labeled with one of the 50 sound classes. This dataset has been widely used in sound classification research and provides a benchmark for evaluating the performance of sound classification models. The ESC-10 dataset is a subset of the ESC-50 dataset with 10 classes.

In this project, we propose a transfer learning approach using YAMNet for environmental sound classification on the ESC-50 and ESC-10 datasets. Specifically, we fine-tune the pre-trained YAMNet model [1] on the ESC-50 and ESC-10 datasets and evaluate the effect of different layer sizes on the performance of the model.

3 Methodology

In our approach, we utilize the pre-trained YAMNet model, which has been trained on the extensive AudioSet dataset, to extract feature embeddings from the audio recordings within the ESC-50 dataset. YAMNet is a convolutional neural network (CNN)-based deep learning model that combines feature extraction from sound signals with a classifier for sound class prediction.

To prepare the audio data for fine-tuning, we perform necessary preprocessing steps and generate output embeddings which are then used as input for our model. This approach enables the model to learn dataset-specific features while retaining the general features acquired from the pre-training phase. To evaluate the performance of the fine-tuned YAMNet model, we consider various metrics such as accuracy, precision, recall, and F1-score.

In our experiments, we investigate the impact of different layer sizes and depths on the performance of the fine-tuned model. Specifically, we utilize Dense layers with varying parameter sizes and depths, employing the ReLu activation function. Finally, we incorporate a softmax layer with either 50 or 10 outputs, depending on whether the ESC-50 (50 classes) or ESC-10 (10 classes) dataset is used.

By leveraging the pre-trained YAMNet model and fine-tuning it with customized layers, we aim to enhance the model's ability to capture the distinctive features of the target datasets, leading to improved classification performance.

Layer	**Neurons**
Input	Embedding(1024)
Dense	128/256/512/1024
Softmax	50/10 (ESC-50/ESC-10)

Table 1: With layer depth 1

Layer	**Neurons**
Input	Embedding(1024)
Dense	128/256/512/1024
Dense	128/256/512/1024
Softmax	50/10 (ESC-50/ESC-10)

Table 2: With layer depth 2

Layer	**Neurons**
Input	Embedding(1024)
Dense	128/256/512/1024
Dense	512
Dense	128/256/512/1024
Softmax	50/10 (ESC-50/ESC-10)

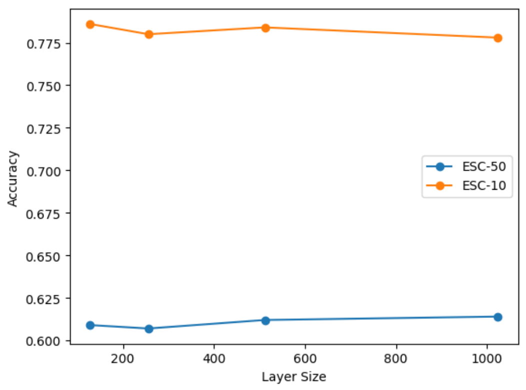
Table 3: With layer depth 3

4 Results

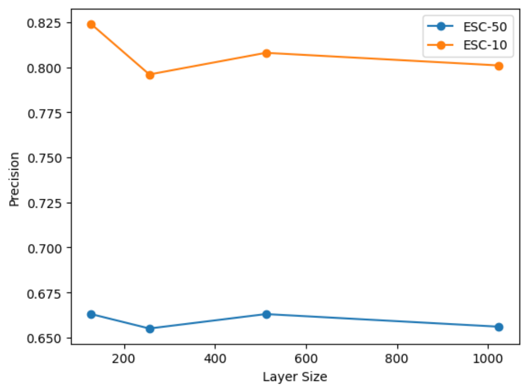
To maintain class balance during the data split, the ESC-50 dataset utilizes a random division into training, validation, and test sets. Each entry in the dataset is labeled with a ”fold number,” aiding in the equitable distribution of classes during the split.

For model training, the Adam optimizer is employed and the models are trained for 20 epochs. Early Stopping is implemented, monitoring the validation loss to determine the best model for the prediction task. To account for the large number of classes, the SparseCategoricalCrossEntropy loss function is utilized.

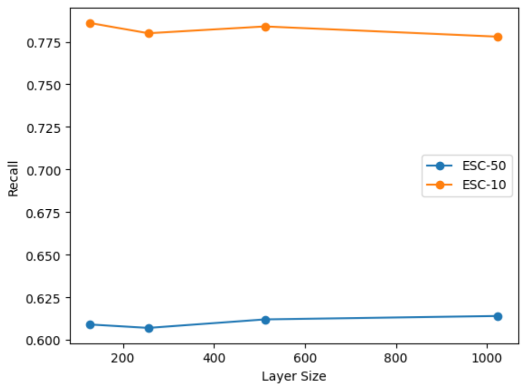
Overall, the dataset is carefully partitioned, an Adam optimizer is used for training over 20 epochs, and Early Stopping based on validation loss ensures the selection of the optimal model. The choice of SparseCategoricalCrossEntropy loss function is well-suited for the task considering the significant number of classes involved.



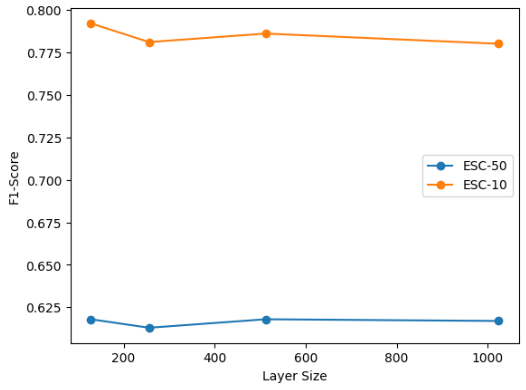
(a) Accuracy



(b) Precision



(c) Recall

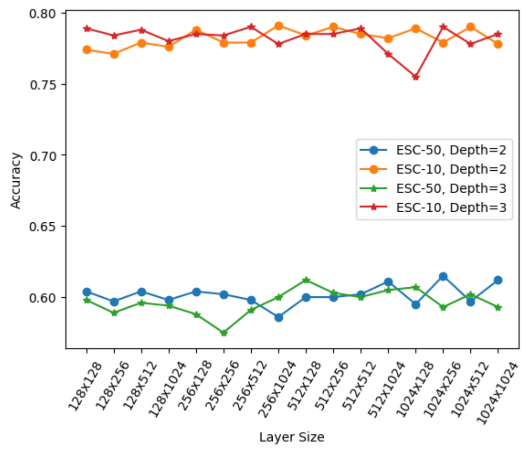


(d) F1-Score

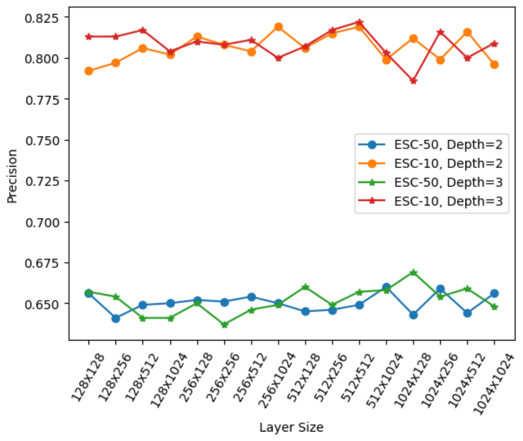
Figure 1: 1 hidden layer

The performance of the models with a hidden depth of 1 was average. ESC-10 outperformed ESC-50, as anticipated since the model had to generalize to 80% fewer classes.

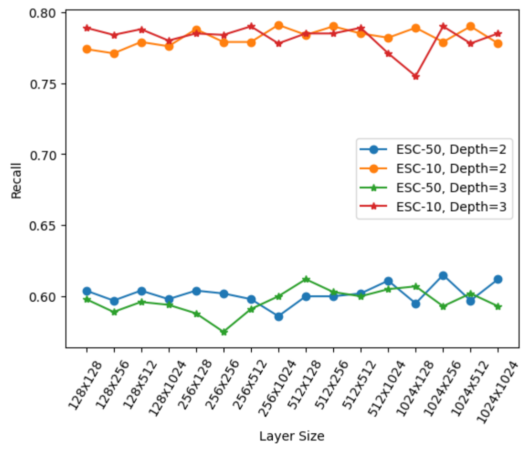
The accuracy values and F1 scores for ESC-50 ranged from 0.6 to 0.625, while for ESC-10, they averaged between 0.775 and 0.8. It is noteworthy that achieving an F1 score of 0.8 is quite good, considering that a simple model with just one layer and 128 hidden units was able to achieve such high scores. This once again highlights the power of transfer learning, where YAMNet, trained on completely different data, produced meaningful embedded representations.



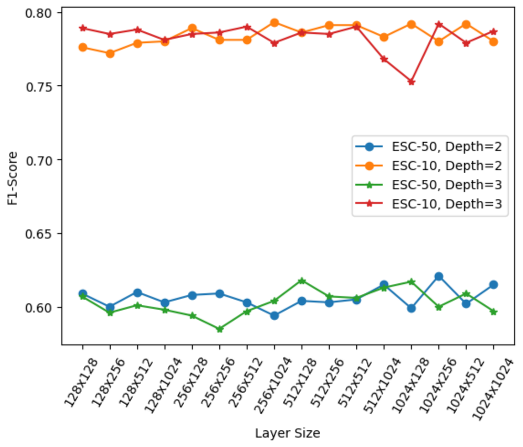
(a) Accuracy



(b) Precision



(c) Recall



(d) F1-Score

Figure 2: Comparison between 2 and 3 hidden layers. For Depth=3, a layer of 512 hidden units is added in between. **Example: 128x128 corresponds to 2 dense layers with 128 and 128 hidden units for Depth=2, corresponds to 3 dense layers with 128, 512, and 128 hidden units for Depth=3.**

Comparable performance was observed on the ESC-10 dataset for both depth 2 and depth 3 models. However, the best-performing models were those with a depth of 3, specifically with 128, 512, and 512 hidden units, achieving accuracy and F1 scores close to 0.8. Nonetheless, depth 2 models were not far behind, indicating that a relatively shallow architecture is sufficient to fine-tune the YAMNet output embeddings.

On the other hand, the models did not exhibit significant improvements over depth 1 when evaluated on the ESC-50 dataset. The limited amount of data available for each class (only 40 examples) is a primary factor contributing to this result. The model struggles to learn effective generalization due to the small dataset size. To address this limitation, future improvements could incorporate data augmentation techniques. For example, introducing noise to the training samples can enhance the model's robustness and generalizability, while also increasing the size of the dataset.

Other improvements for the ESC-50 model that might help would be ensemble methods, combining multiple models to improve overall performance by leveraging diverse representations and reducing individual model biases, but I believe that given enough data the performance on ESC-50 should be comparable to that on ESC-10.

5 References

[1] https://github.com/tensorflow/models/tree/master/research/audioset/yamnet

[2] Shor, Joel, et al. **"Towards Learning a Universal Non-Semantic Representation of Speech."** Interspeech 2020, 2020, pp. 140–44. arXiv.org, https://doi.org/10.21437/Interspeech.2020-1242.

[3] https://github.com/karolpiczak/ESC-50

[4] Zhuang, Fuzhen, et al. **A Comprehensive Survey on Transfer Learning.** arXiv, 23 June 2020. arXiv.org, http://arxiv.org/abs/1911.02685.

[5] https://www.tensorflow.org/tutorials/

[6] https://github.com/ShubhJaroria/Transfer-Learning-YAMNet