COMP9444 Project Summary

Environmental Sound Classification

Hongtian Yu, Zhenhao (Harley) Ding, Zachary Scott, Weiyuan (Felix) Li, Aalam Virk

I. Introduction

Traditionally, sound-based analysis is a vital piece in the decision-making process across many fields and disciplines. However, with the modern mass adoption of technological advancements and a rising desire for automated solutions, the need for robust and efficient artificial audio classification models grows accordingly. The value of these systems are often found to lie in their ability to run with little compute power on embedded devices, leveraging neural networks that can accurately and efficiently identify environment sounds in the devices' surroundings, enabling robotics and other larger systems to better interpret and evaluate sounds to optimise decision making.

The usage of automatic environmental sound classification (ESC) models in embedded devices are widespread across multiple prominent industries. Whilst a large amount of research has already been put into high accuracy models, in many cases the cost of both training and inference are remarkably high. In physical applications such as security systems, robotics, or healthcare, it is found that such models are not optimal in terms of cost and speed despite being able to achieve an extremely high accuracy. In particular, the difference between requirements of such applications and the objectives of aforementioned models mostly lies in the key fact that many real world sound classifiers often do not need to classify a wide, diverse range of sounds with a large number of categories; rather, they only need to be able to distinguish between a small number of kinds of sounds.

Hence, this presents our key problem, to leverage signal processing and deep learning techniques to develop a robust and efficient model capable of running on limited computational size and power, without sacrificing a significant degree of accuracy, and, ideally, has minimal training cost.

The applications of this project are rich and diverse: n security, identifying of sounds such as glass breaking, talking and footsteps[9] to aid law enforcement and home security applications; in wildlife monitoring, detecting animals through audio signals for track cameras for hunting or research into animal populations; in healthcare, through patient monitoring, identifying specific human sounds to alert caretakers and medical professionals. In all these cases the speed and accuracy of these models are vital in ensuring their effectiveness, and these models often only need to recognise a small number of audio classes.

II. Related Work

With the current advances in industries requiring ESC, the requirements for case-specific models are ever increasing. The earlier attempts at ESC models have not adequately addressed the constraints posed by embedded devices whilst simultaneously minimizing training costs.

Mohaimenuzzaman et al. [7] tackled a similar problem in extremely resource constrained devices (micro controllers) through simplifying ACDNet architecture model in a hybrid compression pipeline technique. Using a CNN (convolutional neural networks) architecture followed with batch normalization, reLU and linear layer dropout the network forms two feature extraction blocks aimed to learn high and low level features separately in series. Using this network with ~101 million parameters it is trained on the ESC dataset achieving 96.65% before the compression where they sparsify the model's weight matrices and apply structured compression where this process prunes off insignificant nodes, decreasing the models size by 97.22% and 97.28% fewer flops. This achieves an optimized model Micro-ACDNet with significantly reduced size however maintains a state-of-the-art accuracy of 96.25% allowing the model to run effectively on "extremely resource-constrained devices." The limitation of this technique lies in the significant increase in training time of this technique, as at every compression stage, the model must be re-trained in order to maintain its accuracy, increasing computational training cost. In implementing this paper, they had to perform machine learning on the "cutting edge" which is, unfortunately, not supported by standard frameworks which makes implementing this extremely difficult, and inflexible to pivot to other neural networks. Furthermore, when running the published source code locally we not only found the approach to be unsuitable as stated above, but also that it failed to reach the reported accuracy, solidifying our decision to stay away from compression pipelines.

This recommended baseline for the ESC dataset by its author Piczak[5], aimed to leverage computer vision techniques of convolutional and pooling layers to extract and learn local structures present in an inputted log-mel spectrogram and use these characteristics to classify environment sound. This paper exhibits 3 major areas; data preprocessing, data augmentation and CNN architecture. Starting with data-preprocessing, the paper suggests using a mel-spectrogram (frequencies

converted to a mel scale) of the audio and passing frames of its overlapping segments into the network, which we found to be a common technique among all the papers as learning on whole clips was too limiting due to the number examples available for training. Despite the success with pre-processing, then running a local implementation including data augmentation we found the techniques of applying random time delays, class-dependent time stretching and pitch shifting to be ineffective at boosting model performance only increasing the computational run-time. Finally, despite the 4 layers network split into 2 CNN layers with max pooling and 2 fully connected layers implemented by Piczak seeming too simple for this task, we found that a large number of classes were zero padded meaning the number of features the model could possibly learn on was limited without further signal processing techniques. Thus, our methodology to implement this baseline with better initial augmentation and pre-processing techniques and then expand on the architecture, once we have extracted more or better features from the dataset would be considerably more beneficial than initially improving the model on unprocessed weak data.

A paper on TEO-based gammatone features[6] came to the same conclusion about the dataset and Piczak network, utilizing a near identical baseline model however using a Teager energy operator (TEO) and Gammatone filter bank for pre-processing which it claimed would be better for capturing energy variations in the audio signals provided by the dataset, resulting in more features and better performance. The CNN part of the model in this paper was kept identical however the linear layers were reduced to 1 layer of 500 parameters from the 2 layers with 5000 parameters suggested in the paper above, as it did not influence performance however sped up training time. Using these two techniques the GTSC alone improved the model 12.9% and the TEO-GTSC improved 12.35% on the ESC-50 and Urban sound 8K datasets both harder than ESC-10 where it is likely to have at least as great of an improvement due to it being a simpler dataset. This increases in accuracies alone without increasing model complexity provide us with a foundation to combine and implement the techniques found in this paper with other related works.

The accuracies in these papers were sufficient to surpass the human benchmark of ~80% on ESC-50 dataset even though the papers proposed models capable to work with limited computational resources. Nonetheless, state-of-the-art accuracies were not achieved using these methodologies, giving us a solid foundation for further experimentation.

III. Methods

To meet our problem statement, we favour relatively small architectures over larger, complex ones. To start, we replicated the recommended baseline for ESC-50 and ESSC-10 proposed by Karol Piczak, the author of the ESC datasets, and used it as our baseline model. Piczak CNN has been fine tuned for its task, it contains 2 convolutional layers followed by 3 fully connected layers with dropout. In addition, Piczak's model utilized Log-Mel (which takes the log of the Mel-frequency[1]) for its preprocessing, which provides a more perceptually meaningful representation of the sounds. The relatively small scale of the Piczak network when compared to other models such as VGG-16 aligns with our goal, thus it became our baseline.

We had multiple approaches to improve the baseline model, where:

- Further optimize the architecture of the network
- Further fine-tune the hyper-parameters
- Extend the existing signal pre-processing pipeline
- Explore other deep learning techniques (such as attention models)

We attempted to first optimize the architecture by tweaking the layers and working with the hype-parameters. However, since the original baseline is already finely optimised, any adjustments we made only negatively impact the accuracy and inference speed. Therefore, we focused on how to extend the existing signal pre-processing pipeline such that we can extract more features from the raw audio. In addition, we realized we could incorporate a novel type of parallel CNN mechanism into our network, which on paper would increase accuracy since the network is able to capture both the temporal features with rectangular kernels on one branch as well as the more detailed local features on another, deeper branch using square filters.

The pre-processing pipeline we utilized was inspired by the paper: Novel TEO-based Gammatone Features for Environmental Sound Classification[6], which has been reported seeing an increase in classification accuracy. However, due to the vague descriptions in the paper and lack of published source code, we had to apply a considerable number of our own heuristics as well as improvisation during the replication process, to the extent that our replication thereof should be seen as a semi-unique model in itself.

Ultimately, due to the extra features provided by our signal-preprocessing pipeline, we hypothesised that we would be able to remove one of the fully connected layers while maintaining our initial accuracy. This significantly reduced the number of parameters in the network.

IV. Experimental Setup

ESC dataset[3] is comprised of a collection of 2000 labelled environment recordings of 5 seconds from the collaborative publicly available Freesound project[4]. These recordings are grouped into 5 major categories consisting of 10 classes:

- Animal sounds
- Natural soundscapes and water sounds
- Human (non-speech) sounds
- Interior/domestic sounds
- Exterior/urban noises

Each of these noises are sampled to a unified format *.wav files, single channelled 44.1kHz and manually labelled audio. To ensure consistency the author has manually partitioned the data into 5 folds allowing for 5-fold cross validation which makes sure the benchmarks of models against each other consistent.

As this project is aimed at light weight, efficient and few-class classification models we can select a premade subset of this dataset, ESC-10, comprising of fewer but still diverse soundscapes. The dataset is manually selected by Karol Piczak the author of the ESC dataset where he partitions existing audio in ESC-50 into 3 general categories containing a total of 10 classes and splits the data into predefined 5 folds again.

In our implementation we used a hugging face dataset[5] to provide a streamlined API for data manipulation, as the raw ESC dataset's GitHub includes the waveforms separated to their meta data which is held in a CSV (fold number, target, category, esc10 (boolean, if the entry is in esc10 or not), src_file and take) while the hugging face dataset has precombined them. As a baseline we inherited the hyperparameters of the baseline paper by ESC's author[6] with a selection process based mostly on heuristics. To test these, we implemented his baseline model and fine-tuned the hyperparameters given for the best performance relative to validation accuracy.

Our final learning rate for the peak performance of the baseline model was decided to be a relatively lower learning rate of ~0.002, and we kept the proposed momentum of 0.9 as we found it helped overcoming small local minima and smoothing the model's validation accuracy trajectory; we estimate that these values derive from the high complexity of the ESC dataset with static and irregular features between samples. As the dataset is quite complex and intricate, we kept the proposed 0.001 value for L2 weight decay, and 0.5 dropout probability given for the fully-connected layers and first convolutional layer as we expected a large amount of overfitting to be done by the model, particularly without augmentation.

V. Results

Our model and all other models tested were evaluated by the standard ESC metric of 5-fold cross validation regime where at each fold, the 4 other folds are used for training and the current for validation accuracy. The accuracy reported below varies from either the model hitting the epoch limit of selected from analysis of their accuracy on single fold validation or reaching 100% training accuracy.

Model	ESC-10 Accuracy (avg) ± SE
VGG-16 + Log-Mel	61.75% ± 1.823
TEO-CNN + TEO-GTSC	81.75% ± 3.599
Piczak CNN + Log-Mel	82.75% ± 2.559
Piczak CNN + TEO-GTSC (pure)	82.00% ± 2.885
Piczak CNN + TEO-GTSC (fused)	85.50% ± 3.174
OurNet + Log-Mel	89.50% ± 1.891
OurNet + TEO-GTSC (pure)	88.75% ± 2.372
OurNet + TEO-GTSC (fused)	86.25% ± 3.000

Table 1: validation accuracy results and standard error

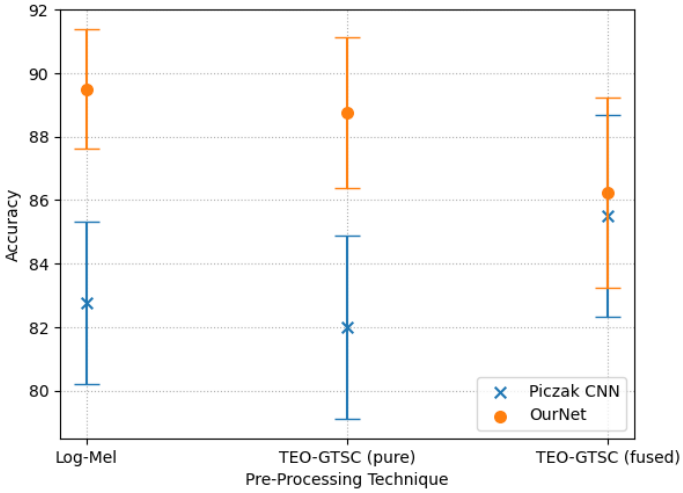

Figure 1: plotted accuracies of models in table 1

From table, the deep CNN architecture VGG-16 and the shallow CNN proposed in the TEO paper these models give an upper and lower bound on the classification accuracy. In each of these scenarios the architecture is either too deep leaving it susceptible to excessive overfitting or too shallow capture all the features present in the input data.

As seen in Figure 1, the Piczak network achieves the higher overall validation accuracy using the fused TEO-GTSC[5]. This is expected TEO-GTSC [7] greatly elevated the extractable features and essentially eliminates low frequency zones, which is advantageous towards deep Convnets. Piczak also achieved comparable accuracy between pure TEO-GTSC and Log-mel. Comparing Piczak's performance on fused TEO-GTSC spectrograms to OurNet's, we hypothesise that due to the decreased linear layers of OurNet it has issues differentiating between features captured by the convolutional layers. With preprocessing techniques, we estimate that there is a balance point between the

style and quantity of information gained by data pre-processing techniques and with the depth of the network's linear layers. In general log-mel loses information within the lower frequency areas which instead are picked up within the signal processing steps of the TEO and Gammatone filter bank technique. This causes Piczak with Log-Mel to neglect high-frequency features, hence losing significant information on training and therefore having a 7.5% difference to our network. We accounted for the lack of high frequency feature by introducing a second parallel CNN which picks up higher frequency features with deeper convolutions and then concatenating them together. And as mentioned in the reasoning with Piczak TEO-GTSC before, the shallowness of our network greatly reduces overgeneralization and ultimately allow it to achieve better accuracies and speed than Piczak TEO-GTSC, whilst not being either overly deep like ACDNet or using large pretrained network like ESResnet to achieve reasonable accuracies.

VI. Conclusions

The goal of this paper was to propose a deep learning architecture leveraging data augmentation and preprocessing techniques capable of achieving state of the art performance on a small number of classes with minimal training cost. The proposed solution to this issue has numerous advantages; exceptional training speed, small number of trainable parameters and state of the art accuracy for its "weight class". However, these don't come without drawbacks, the model is unreliable between folds where the OurNet varies between greatly between folds ranging between 92.5% on the simpler folds and 81.25% on the more complex folds. A further shortcoming is that due to the size of a smaller network as per the problem constraints it falls short of the top of the range accuracy models due to their dramatically increased size and computational power required to train and use them, such as MSClap[11] and Transformer models[12].

With extra time for refining the model, time intensive processes such as the ACDNet structured compression may yield a dramatically smaller and faster version of OurNet. However, this process as stated before is heavily time intensive training the model repeatedly at each reduction in size during the pipeline. A known better option for our pretraining is to use an image captioning or generation network (i.e. ResNet50 [10], RGAN, etc), and to have the encoder feed into our custom network. One possible downside of this improvement to accuracy is that it will greatly increase the memory overhead requirement for this model which invalidates our starting point of edge computing. However, this can be reduced by the ACDNet pipeline described earlier to potentially refine OurNet to be smaller, more efficient and have a greater accuracy. In conclusion, OurNet with current capabilities provides a satisfactory solution to the given problem.

VII. References

[1] - Roberts, L. (2024) *Understanding the mel spectrogram, Medium*. Available at: https://medium.com/analytics-vidhya/understanding-the-mel-spectrogram-fca2afa2ce53 (Accessed: 17 July 2024).

[2] - Piczak, K. (2015) *Karolpiczak/ESC-50: ESC-50: Dataset for environmental sound classification, GitHub*. Available at: https://github.com/karolpiczak/ESC-50 (Accessed: 06 July 2024).

[3] - Font, F., Roma, G. and Serra, X. (2013) 'Freesound technical demo', *Proceedings of the 21st ACM international conference on Multimedia* [Preprint]. doi:10.1145/2502081.2502245.

[4] - Ashraq, I. (no date) *Ashraq/ESC50 · datasets at hugging face, ashraq/esc50 · Datasets at Hugging Face*. Available at: https://huggingface.co/datasets/ashraq/esc50 (Accessed: 06 July 2024).

[5] - Piczak, K. (no date) *Environmental sound classification with ..., ENVIRONMENTAL SOUND CLASSIFICATION WITH CONVOLUTIONAL NEURAL NETWORKS*. Available at: https://www.karolpiczak.com/papers/Piczak2015-ESC-ConvNet.pdf (Accessed: 07 July 2024).

[6] - Agrawal, D.M. *et al.* (2017) 'Novel teo-based Gammatone features for Environmental Sound Classification', *2017 25th European Signal Processing Conference (EUSIPCO)* [Preprint]. doi:10.23919/eusipco.2017.8081521.

[7] - Mohaimenuzzaman , M. *et al.* (2022) *Environmental sound classification on the edge: A pipeline for deep acoustic networks on extremely resource-constrained devices, Pattern Recognition*. Available at: https://www.sciencedirect.com/science/article/abs/pii/S0031320322005052?via%3Dihub (Accessed: 10 July 2024).

[8] - Fang, Z. *et al.* (2022) 'Fast environmental sound classification based on resource adaptive Convolutional Neural Network', *Scientific Reports*, 12(1). doi:10.1038/s41598-022-10382-x.

[9] - Bansal, A. and Garg, N.K. (2022) 'Environmental sound classification: A descriptive review of the literature', *Intelligent Systems with Applications*, 16, p. 200115. doi:10.1016/j.iswa.2022.200115.

[10] - Guzhov, A. *et al.* (2020) *ESResNet: Environmental sound classification based on visual domain models, arXiv.org*. Available at: https://arxiv.org/abs/2004.07301 (Accessed: 01 August 2024).

[11] - Elizalde, B., Deshmukh, S. and Wang, H. (2024) *Natural language supervision for general-purpose audio representations*, *arXiv.org*. Available at: https://arxiv.org/abs/2309.05767 (Accessed: 05 August 2024).

[12] - Liu, X. *et al.* (no date) *ArXiv:2303.07626v1 [CS.SD] 14 Mar 2023, CAT: CAUSAL AUDIO TRANSFORMER FOR AUDIO CLASSIFICATION*. Available at: https://arxiv.org/pdf/2303.07626 (Accessed: 05 August 2024).