ENSEMBLE OF CONVOLUTIONAL NEURAL NETWORKS FOR THE DCASE 2020 ACOUSTIC SCENE CLASSIFICATION CHALLENGE

Technical Report

Paulo Lopez-Meyer[1]*, Juan A. del Hoyo Ontiveros*[1]*,*
Georg Stemmer[3]*, Lama Nachman*[2]*, Jonathan Huang*[4]

[1] Intel Corp, Intel Labs, Av. Del Bosque 1001, Zapopan, JAL, 45019, Mexico,
{paulo.lopez.meyer, juan.antonio.del.hoyo.ontiveros}@intel.com
[2] Intel Corp, Intel Labs, 2200 Mission College Blvd., Santa Clara, CA 95054, USA,
{lama.nachman}@intel.com
[3] Intel Corp, Intel Labs, Lilienthalstrasse 15, 85579, Neubiberg, Germany,
georg.stemmer@intel.com
[4] Work done at Intel, jonathan.huang@ieee.org

ABSTRACT

For the DCASE 2020 Task 1a, we propose the use of three different deep learning based convolutional neural networks architectures: AclNet, AclResNet50, and Vgg12. These three neural network architectures were pre-trained with Audioset data for embedding generation, and then fine-tuned with an added classification layer, through the development dataset provided by the task. The outputs produced by these trained models proved to be complementary when ensembled, due to the different nature of the feature front-end, and of architecture diversity. The ensemble average of these models' outputs improved significantly from best single model classification accuracy of 67.55% to 69.74% on the evaluation dataset, when trained with the challenge suggested development partitioning.

Index Terms— Acoustic Scene Classification, Deep Learning, Convolutional Neural Networks, Transfer Learning, End-to-End Audio Classification, Ensemble Averaging

1. INTRODUCTION

For the 2020 Detection and Classification of Acoustic Scenes and Events challenge (DCASE2020), acoustic data were provided to solve different audio related tasks. Task 1 refers to the challenge of building a model to classify different recordings into predefined classes corresponding to different urban environment scenes.

Following the guidelines provided by the challenge in the Task1 subtask a (Task 1a), we experimented with three different deep learning (DL) convolutional neural network architectures (CNN): AclNet, AclResNet50, and Vgg12. AclNet and AclResNet 50 are two end-to-end (e2e) architectures that take raw audio data as the input into two 1D convolutional layers, followed by a 2D multi-layer CNN, i.e. a VGG type and a ResNet type, respectively. On the other hand, the Vgg12 architecture is based on well known computer vision 12-layer CNNs adapted for audio classification; this CNN model takes Log-Mel filterbanks of 64 spectral dimensions as input features.

2. METHODOLOGY

All the implementations and experimentation performed in our work submitted to the DCASE2020 challenge Task 1a are explained in detail in the following sections.

2.1. Data Processing

The DCASE2020 Task 1a dataset consists of 10-second audio recordings obtained at 10 different acoustic scenes: airport, indoor shopping mall, metro station, pedestrian street, public square, street with medium level of traffic, traveling by tram, traveling by bus, traveling by and underground metro, and urban park , recorded at 12 major European cities [1].

The challenge suggests the usage of a 1-fold arrangement for development as part of this task, i.e. 13,962 audio samples for training, and 2,968 for evaluation. Through the development stage of our implementations, we used Google Audioset data [2] to construct efficient audio embedding generators customized for each one of our implemented DL architectures.

The dataset for this task comprises single-channel audio recordings at 44.1 kHz of sampling rate in 24 bit resolution. For the development of the two e2e CNN architectures, all audio signals were downsampled from its original sampling rate to 16 kHz. For the development of the Vgg12 spectral based CNN architecture, the audio data were processed to generate Log-Mel filterbank representations with 64 filter bands over a time window of 25 milliseconds and overlaps of 10 milliseconds, resulting in one Log-Mel filterbank channel.

2.2. Convolutional Neural Networks for Acoustic Scene Classification

2.2.1. AclNet and AclResNet50

AclNet and AclResNet50 are e2e CNN architectures which take raw time-domain input waveform, as opposed to more commonly used spectral features, e.g. Log-Mel filterbank or Mel-frequency cepstral coefficients (MFCC). One of the advantages of these types of e2e architectures is that the front-end feature makes no assumptions of

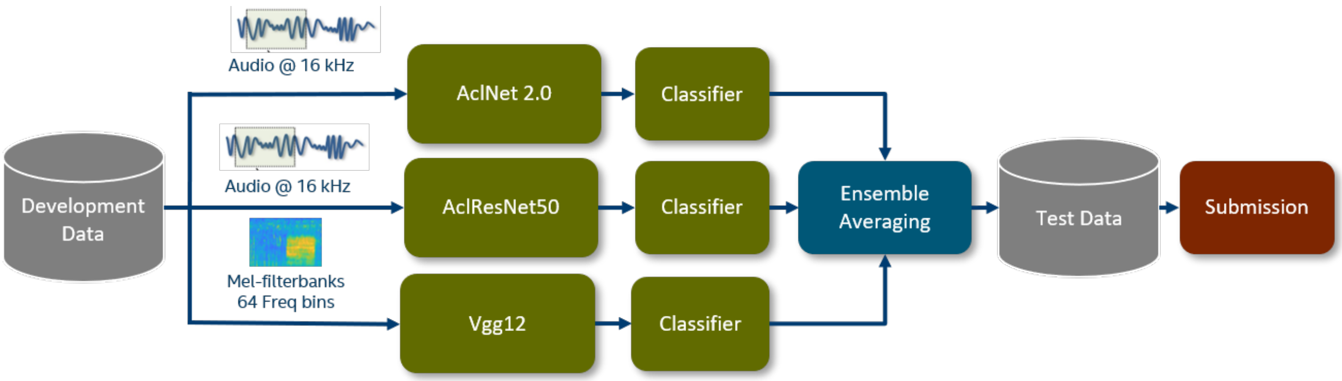

Figure 1: Development of our proposed implementation for four low-memory CNN architectures for audio scene classification in the DCASE202 Task 1b.

the frequency response; its feature representation is learned in a data-driven manner, thus its features are optimized for the task at hand provided there are sufficient training data.

For AclNet, we conditioned the settings corresponding to the work described in [3], with a width multplier of 2.0, and conventional depth-wise convolution layers. The AclNet architecture we developed for the DCASE2020 was pre-trained with Audioset to generate a vector of 512 audio embeddings sent into a fully-connected layer classifier with ReLU activation functions in a transfer learning manner. Raw audio data at 16 kHz from the Task 1a dataset were fed to the pre-trained AclNet, and embeddings were used to train the classifier.

For AclResNet50, we replaced the CNN component of AclNet after the two input 1D convolutional layers, with a ResNet type CNN commonly used in image recognition [4], comprising 50 convolutional layers. As with AclNet, we pre-trained this architecture with Audioset for audio embeddings generation, and performed transfer learning to train an added fully-connected layer with ReLU activation functions to perform the final classification.

In order to increase the robustness of the training process, we also used different data augmentation techniques commonly used in audio processing such as random noise addition, random cropping of 1-second of the audio signal, and random gain variation, together with the widely used mixup data augmentation technique [5]. During the training, audio data were randomly selected from mini-batches of training clips. At evaluation time, we run the inference on 1-second non-overlapping consecutive audio segments, and then averaged the outputs over the length of the evaluation audio.

2.2.2. *Vgg12*

The Vgg12 model used in our proposed work is an adaptation of the well-known VGG architecture [6], that has proved to be an efficient approach for audio classification. It has a total of 12 convolutional layers, with the first one having an output of 64 channels, and the last one is defined by 512 outputs. At the output of each convolutional layer, we apply batch normalization followed by ReLU activation. Through the convolutional layers, there are 5 max-pool layers with kernel size of 2. This CNN architecture is designed for variable input size (e.g. 64 spectral dimensions and arbitrary number of time steps). The output of the last convolutional layer is average pooled, to always produce a vector length of 512 values. This vector is then followed up by a fully connected layer to produce the 10-class output defined by the challenge's Task 1a.

During the training phase, 5-seconds of audio are randomly selected from the training clip. Spec Augment [7] was used as a data augmentation process that proved beneficial in our experiments, by randomly placing two masks over each time and frequency axis of random width between 0-25% of the Log-Mel filterbanks input width and height. We also tried mixup augmentation, but did not observed any benefit from using it on our Vgg12 experimentation. At evaluation time, we run the inference on 1-second non-overlapping segments, and then average the outputs over the length of the evaluation audio.

2.3. Training Strategy

In order to have an efficient training, we performed a search for the optimal parameters of these audio classification CNNs . We experimented with different values and configurations, that yield to the best performing models; the best training hyper parameters found (learning rate LR, learning rate decay LR-d, number of epochs where the best model was found E, weight decay WD, and drop out rate DO) of each of the three architectures described in the previous subsection are listed in Table 1.

All three CNN architectures were trained with the Adam optimizer. During the fine-tuning process, we kept the part of the corresponding Audioset pre-trained network with a LR value that is 1/10th of the LR as the rest of the network. The evaluation dataset was used for model selection, i.e. the best performing model on the evaluation dataset was saved and used for validation inference. Pytorch was the framework used in our work to build the CNNs described.

2.4. Ensemble of Convolutional Neural Networks for Audio Scene Classification

In order to reduce individual variance of each of the developed CNN models, e.g. AclNet, AclResNet50, and Vgg12, we applied a simple ensemble averaging technique, commonly used in machine learning to improve the prediction performance [8]. This approach basically and simply consists on the averaging of the prediction scores obtained by different models, i.e. average of the ten softmax output scores across the three different models.

By combining the prediction scores from different CNN models, the ensemble yields to results above the reported challenge baseline (54.1%); the intention is to add a bias that counters the variance of an individually trained model. Having a diversity of

Table 1: Training setup values for the three CNN architectures proposed. These values are the learning rate, learning rate decay, epoch of best model obtained, weight decay, and drop out rate, respectively.

Architecture	LR	LR-d	E	WD	DO
AclNet	1e-4	0.01	150	2e-4	0.25
AclResNet50	1e-4	0.001	5	1e-6	0.20
Vgg12	1e-3	0.01	131	1e-6	0.90

Table 2: Best classification results over the evaluation set obtained by each individual CNN architectures explored in this work, and the ensemble averaging of these three.

CNN Architecture	Trainable Parameterss	Accuracy
Baseline	***	54.10%
AclNet	2.7M	65.53%
AclResnet50	24.6M	65.46%
Vgg12	12.6M	67.55%
Ensemble	39.9M	68.77%

CNN models helps to achieve this intention. The experimental results obtained for the most obvious combination, i.e. ensemble of the three CNN described in this work, are shown in the next section.

3. RESULTS

The experimental results obtained from our three developed CNN models for acoustic scene classification over the evaluation dataset are shown in Table 2 and Table 3. The number of trainable parameters is also listed, in order to present an context of the size of the architectures experimented with. During experimentation, we noticed that performing an ensemble of the best individual models does not guarantee to yield into the highest ensemble results. Because of this, we did two experiments; the first one consists on the ensemble the outputs of the best individual models (see Table 2); the second one consists on finding the best combination of individual models that yields into the highest ensemble accuracy (see Table 3). The experimental results presented here represent the best evaluation accuracy performance achieved by our CNN models and the ensemble averaging of them, which constitutes our submission for Task 1a, at the time of the DCASE2020 submission deadline.

Table 3: Classification results over the evaluation set obtained by each three individual CNN architectures that yield to the highest ensemble accuracy found.

CNN Architecture	Trainable Parameterss	Accuracy
Baseline	***	54.10%
AclNet	2.7M	65.53%
AclResnet50	24.6M	64.86%
Vgg12	12.6M	66.41%
Ensemble	39.9M	69.74%

4. CONCLUSIONS

For this year's DCASE2020 Task 1a, we experimented with the ensemble of three different CNN architectures trained for acoustic scene classification; with this approach, we were able to achieve above the baseline results, as reported in the challenge guidelines. The best accuracy results over the evaluation dataset by an individual CNN was 67.55%; our ensemble approach results in a 39.9M parameters model that achieves an evaluation accuracy of 69.74%.

5. REFERENCES

[1] T. Heittola, A. Mesaros, and T. Virtanen, "Acoustic scene classification in dcase 2020 challenge: generalization across devices and low complexity solutions," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2020 Workshop (DCASE2020)*, 2020, submitted. [Online]. Available: https://arxiv.org/abs/2005.14623

[2] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *Proc. IEEE ICASSP 2017*, New Orleans, LA, 2017.

[3] J. J. Huang and J. J. A. Leanos, "Aclnet: efficient end-to-end audio classification CNN," *CoRR*, vol. abs/1811.06669, 2018. [Online]. Available: http://arxiv.org/abs/1811.06669

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015. [Online]. Available: http://arxiv.org/abs/1512.03385

[5] H. Zhang, M. Cissé, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *CoRR*, vol. abs/1710.09412, 2017. [Online]. Available: http://arxiv.org/abs/1710.09412

[6] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, 2015.

[7] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," in *INTERSPEECH*, 2019.

[8] J. Brownlee, "Ensemble learning methods for deep learning neural networks," 2018. [Online]. Available: https://machinelearningmastery.com/ensemble-methods-for-deep-learning-neural-networks