END2END CNN-BASED LOW-COMPLEXITY ACOUSTIC SCENE CLASSIFICATION

Arshdeep Singh[1]*, Dhanunjaya Varma Devalraju*[2]*, Padmanabhan Rajan*[3]

Indian Institute of Technology, Mandi, India
Email: {$d16006^1, S18023^2$}@students.iitmandi.ac.in, [3]padman@iitmandi.ac.in

ABSTRACT

This technical report describes the IITMandi_AudioTeam's submission for ASC Task 1, Subtask B of DCASE2020 challenge. This report aims to design low-complexity systems for acoustic scene classification. We propose a convolution neural network based end-to-end classification framework. The proposed framework learns from raw audio directly. We present performance analysis of various frameworks with model size lesser than 500KB for classification. The three acoustic scenes namely indoor, outdoor and transportation are considered. Our experimental analysis shows that the proposed end-to-end framework, where features are being learned from raw audio directly, with a model size of approx. 77KB gives similar performance on development dataset as that of baseline[1] system proposed for the same task.

Index Terms— Acoustic scene classification, Low-complexity, Convolution neural network.

1. INTRODUCTION

Acoustic scene classification (ASC) aims to classify surrounding physical environment into pre-defined categories using sound information. There exist many real time applications such as context aware services, home surveillance etc. [1]. The ASC systems can be used on portable devices, which has limited storage. In this regard, detection and classification of acoustic scenes and events (DCASE) 2020 challenge aims to come up with low-complexity solutions with a memory constraint of less than 500KB. This report aims to target the DCASE ASC Subtask B [2].

Traditionally, the features inspired form speech and music processing tasks such as time-frequency representations, mel cepstral coefficients, constant-Q-transform are being utilized in acoustic scene classification [3]. Since, the characteristic of acoustic signal produced in the surrounding is very different from speech and music signal in terms of wider pitch, multiple unknown sound sources with varying characteristics, unstructured signal etc. This requires adaptive feature representations to cope up the complexity and variability of acoustic signals in ASC. In this regard, many of the works target on feature learning, which aims to learn from the low-level representations obtained using time-frequency representations [4, 5, 6]. Some of the studies also employ to use transfer-learning based representations [7, 8].

In this report, our aim is to learn representations using raw audio directly. This gives advantage in two ways; first, the feature representations can be adaptively learned from the raw audio itself, with an end-to-end classification framework. Second, the complexity in feature representation and classification can be reduced. Therefore, we propose an end-to-end convolution neural network (CNN) based classification framework. The performance is analyzed by varying the complexity or model size of the proposed framework.

The rest of the report is organized as follows. In section 2, the proposed framework is explained. Performance analysis and conclusions are included in section 3 and 5 respectively.

[1]Baseline uses Log-mel band energies as features and 2-layer CNN with 1 fully-connected layer for classification.

2. PROPOSED METHODOLOGY

In this section, we describe various end-to-end CNN-based architectures designed for experimentation, training and evaluation procedure.

2.1. Various CNN models

Model (A): CNN-205KB

CNN-205KB architecture is a 1D-CNN with model size 204.9KB[2]. The architecture is shown in Figure 1(a). The model consists of three convolution layers, a fully-connected layer (FC) and a classification layer with 3-units having softmax activation function. Each of the convolution layer is followed by batch normalization layer (BN), activation layer using ReLU and a pooling layer either locally average pooling or global average pooling. The number of filters, length of each filter, pooling size and number of units in FC-layer are mentioned in the Figure 1(a). The CNN-205KB has 52467 total and 52467 non-zero parameters number of parameters[2].

Model (B): CNN-73KB

CNN-73KB architecture is a 1D-CNN with model size 72.7KB. The architecture is shown in Figure 1(b). There are two convolution layers which are similar to the first two layers of a network, SoundNet [8][3]. This is followed by global average pooling, a FC-layer and a classification layer. The initial weights of the convolution layers of CNN-73KB model are taken from pre-trained weights of first two layers of SoundNet. The CNN-205KB has 18611 total and 18611 non-zero parameters number of parameters.

Model (C): CNN-77KB

CNN-77KB architecture is a 1D-CNN with model size 77.2KB. CNN-77KB has similar architecture to that of CNN-77KB except the FC-layer has 64 units. The architecture is shown in Figure 1(c). The CNN-205KB has 19763 total and 19763 non-zero parameters.

[2]CNN model size, number of total and non-zero parameters are computed using the script *model_size_calculation.py* as given in DCASE 2020 challenge.

[3]SoundNet is a pre-trained, 1D-CNN, which is trained using transfer-learning from 2 million videos.

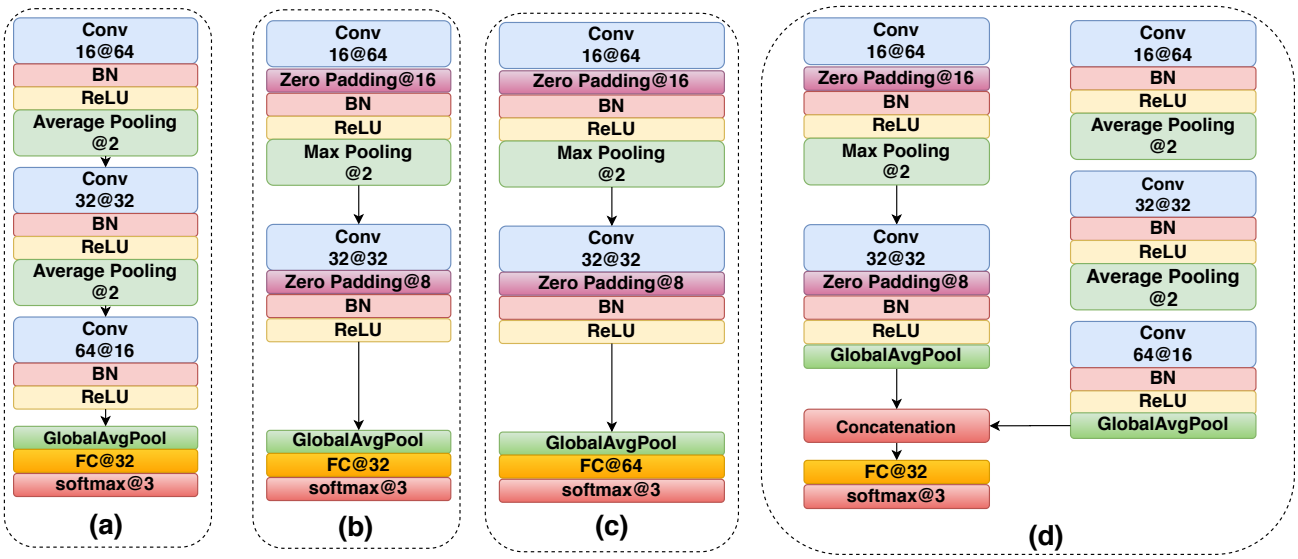

Figure 1: Various architecture details (a) model 1, (b) model 2, (c) model 3 and (d) model 4. The architectures (a), (b), (c) and (d) has a memory size 204.9KB, 72.7KB, 77.2KB and 277.1KB respectively. Here, BN, ReLU, Conv, GlobalAvgPool and FC represents batch normalization layer, rectified linear unit activation function, convolution operation, global average pooling operation and fully connected layer respectively.

Model (D): CNN-277KB

The CNN-277KB has a model size 277.1KB. The model is obtained by concatenating the embeddings obtained from global average pooling layer of *Model A* and *Model C*, followed by a FC-layer with 32-units and a classification layer. The CNN-277KB has 70947 total and 70947 non-zero parameters number of parameters[2].

2.2. Training and evaluation procedure

A given audio recording is divided into M smaller non-overlapping segments, $\{x_1, x_2, ..., x_M\}$. Each segment $x_i \in \mathbb{R}^d$ is considered as a training instance. This gives a training data matrix $\in \mathbb{R}^{d \times T}$, with total number of training instances, $T = N \times M$, each of size d. Here, N indicates total number of audio examples for training.

The CNN model is trained with T-segments. During evaluation, the probabilities obtained from M-segments are averaged to obtained final scores. The ultimate class of a test audio recording is chosen corresponding to the class having maximum final score.

3. PERFORMANCE ANALYSIS

In this section, dataset used for evaluation, training, validation setup and performance analysis of the proposed framework is described.

3.1. Dataset used and Experimental setup

DCASE2020 Task 1 Baseline, Subtask B fold1 development dataset is used for evaluation. The dataset consists of audio recordings of 10s sampled at 48kHz from three scene classes namely indoor, outdoor and transportation. The total number of training and testing examples are 9185 and 4185 respectively.

An audio recording is downsampled to 16kHz and divided into non-overlapping 80-segments (M=80). The gives a total of 734800 segments. We randomly select 80% of segments for training, hence, training data matrix $\in \mathbb{R}^{2000 \times 587840}$. The rest of the 146960 segments are used for validation. Each of the model (A)-(D) is trained for 50 epochs using Adam optimizer. Accuracy and Log loss metric are used for performance analysis.

Initial weights for model (A)-(D)

- Model (A): All the parameters are randomly initialized and updated in training.
- Model (B): The parameters of both convolution layers are initialized with pre-trained parameters of SoundNet. The parameters of fully-connected layer are initialized randomly. All the parameters are updated in training.
- Model (C): The parameters of both convolution layers are initialized with pre-trained parameters from first two layers of SoundNet. The parameters of fully-connected layer are initialized randomly. All the parameters are updated in training.
- Model (D): The parameters of all layers except fully-connected layers are taken from the trained parameters of model (A) and model (C). The model (A) and model (C) are trained using the training data matrix as explained above. During training of the Model (D), only the parameters of fully-connected layer are updated.

The trained models and the relveant codes can be found at given link[4].

3.2. Results

Table 1 shows performance analysis of various proposed CNN architectures as explained in Section 2.1 and baseline model. The baseline model uses Log-mel energies as features and CNN with

[4] https://cloud.iitmandi.ac.in/d/bd487a2974f24e6fabf1/

Table 1: Performance analysis of DCASE 2020 Task 1, Subtask B development test dataset using various proposed CNN architectures as explained in Section 2.1. The performance of baseline framework[2] is also mentioned.

		Various classification models				
		Baseline	Model (A) CNN-205KB	Model (B) CNN-73KB	Model (C) CNN-77KB	Model (D) CNN-277KB
Model size[2]		450.1KB	204.9KB	72.7KB	77.2KB	277.1KB
Class-wise Accuracy	Indoor	82%	77.6%	81.4%	78.8%	87.4%
	Outdoor	88.5%	83.5%	83.8%	89.1%	82.42%
	Transportation	91.5%	93.9%	93.1%	91.9%	92.8%
Class-wise Log loss	Indoor	0.680	0.626	0.579	0.586	0.463
	Outdoor	0.365	0.396	0.383	0.331	0.397
	Transportation	0.282	0.250	0.295	0.298	0.270
Average Accuracy		87.3%	84.9%	85.9%	86.8%	87.2%
Average Log loss		0.437	0.422	0.416	0.399	0.378

2 convolution layers and 1 fully-connected layer for classification. As compared to baseline, our proposed architectures are end-to-end. This means there is no extra computation and memory cost of feature representation. The performance obtained using CNN-77KB, approximately similar to that of baseline. Moreover, the CNN-77KB has approx. 83% lesser model size as compared to that of baseline. The models (B) and (C), which uses pre-train information to initialize the parameters and have fewer parameters, gives better performance as compared to the model (A), where parameters are initialized randomly. Model (D), where embeddings obtained from trained model (A) and model (B) are used to learn classifier, gives better performance than the individual models.

4. CHALLENGE SUBMISSION

We submit four results obtained using the four models (A)-(D) as a final submission for evaluation dataset. The following filenames are used in the submission.

1. Singh_IITMandi_task1b_1 : Predictions generated by Model (A).
2. Singh_IITMandi_task1b_2 : Predictions generated by Model (B).
3. Singh_IITMandi_task1b_3 : Predictions generated by Model (C).
4. Singh_IITMandi_task1b_4 : Predictions generated by Model (D).

5. CONCLUSION

This report focuses on low-complexity system for acoustic scene classification. We propose an end-to-end convolution neural network framework, which learns from the raw audio directly. The proposed framework provides similar results as that obtained using commonly used Log-mel features, but with low-complexity, in terms of memory and feature computation.

6. REFERENCES

[1] D. Barchiesi, D. Giannoulis, D. Stowell, and M. D. Plumbley, "Acoustic scene classification: Classifying environments from the sounds they produce," *IEEE Signal Processing Magazine*, vol. 32, no. 3, pp. 16–34, 2015.

[2] A. Mesaros, T. Heittola, and T. Virtanen, "A multi-device dataset for urban acoustic scene classification," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2018 Workshop (DCASE2018)*, November 2018, pp. 9–13. [Online]. Available: https://arxiv.org/abs/1807.09840

[3] T. Virtanen, M. D. Plumbley, and D. Ellis, *Computational analysis of sound scenes and events*. Springer, 2018.

[4] A. Rakotomamonjy and G. Gasso, "Histogram of gradients of time-frequency representations for audio scene classification," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 23, no. 1, pp. 142–153, 2015.

[5] V. Bisot, R. Serizel, S. Essid, and G. Richard, "Acoustic scene classification with matrix factorization for unsupervised feature learning," in *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2016, pp. 6445–6449.

[6] A. Singh, A. Thakur, and P. Rajan, "Ape: Archetypal-prototypal embeddings for audio classification," in *2018 IEEE 28th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 2018, pp. 1–6.

[7] A. Singh, A. Thakur, P. Rajan, and A. Bhavsar, "A layer-wise score level ensemble framework for acoustic scene classification," in *2018 26th European Signal Processing Conference (EUSIPCO)*, 2018, pp. 842–846.

[8] Y. Aytar, C. Vondrick, and A. Torralba, "Soundnet: Learning sound representations from unlabeled video," in *Advances in Neural Information Processing Systems*, 2016, pp. 892–900.