LOW-COMPLEXITY CNNS FOR ACOUSTIC SCENE CLASSIFICATION

Arshdeep Singh, Mark D. Plumbley

Centre for Vision, Speech and Signal Processing (CVSSP)
University of Surrey, UK
Email: $\{arshdeep.singh, m.plumbley\}$@surrey.ac.uk

ABSTRACT

This paper presents a low-complexity framework for acoustic scene classification (ASC). Most of the frameworks designed for ASC use convolutional neural networks (CNNs) due to their learning ability and improved performance compared to hand-engineered features. However, CNNs are resource hungry due to their large size and high computational complexity. Therefore, CNNs are difficult to deploy on resource constrained devices. This paper addresses the problem of reducing the computational complexity and memory requirement in CNNs. We propose a low-complexity CNN architecture, and apply pruning and quantization to further reduce the parameters and memory. We then propose an ensemble framework that combines various low-complexity CNNs to improve the overall performance. An experimental evaluation of the proposed framework is performed on the publicly available DCASE 2022 Task 1 that focuses on ASC. The proposed ensemble framework has approximately 60K parameters, requires 19M multiply-accumulate operations and improves the performance by approximately 2-4 percentage points compared to the DCASE 2022 Task 1 baseline network.

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

1. INTRODUCTION

Convolutional neural networks (CNNs) have shown state-of-the-art performance in comparison to traditional hand-crafted methods in various domains [1]. However, CNNs are resource hungry due to their large size and computational complexity [2, 3], and hence it is difficult to deploy CNNs on resource constrained devices. For example, Cortex-M4 devices (STM32L496@80MHz or Arduino Nano 33@64MHz) have a maximum allowed limit of 128K parameters and 30M multiply-accumulate operations (MACs) per second during inference. Thus, the issue of reducing the size and the computational cost of CNNs has drawn a significant amount of attention in the detection and classification of acoustic scenes and events (DCASE) research community.

In the literature, some CNN parameters such as filters or weights may be redundant, and contribute to extra memory and computational complexity only [4, 5]. For example, Li et al. [6] found that 64% of the parameters which contribute approximately 34% of computation time are redundant. Removing the redundant parameters from CNNs gives similar performance with an advantage of reduced memory and less computational cost.

The majority of the methods applied to eliminate redundant parameters are on filter pruning [7, 8, 9, 10], where a redundant filter from the network is being eliminated. Filter pruning methods have been widely employed in the computer vision. However, only a few works [5, 11] have applied filter pruning methods in the audio domain, and the issue of designing CNNs for resource constrained devices with constraints on both memory and MACs have not yet been fully explored. Recently, the DCASE challenge Task 1 focuses on designing low-complexity frameworks for ASC with constraints on both memory and MACs. The DCASE challenge 2022 Task 1 [12] provides a baseline CNN, which has 46512 parameters and 29.24M MACs, and achieves an ASC accuracy and log-loss approximately 43% and 1.575 respectively.

This paper aims to design "low-complexity CNNs" which have a maximum number of parameters less than 128K and a maximum number of MACs per seconds less than 30M and performance better than that of the DCASE 2022 Task 1 baseline network for ASC.

The major contributions of the paper is summarized below,

(a) We design a "low-complexity" CNN that has fewer parameters, fewer MACs and better classification performance than the DCASE 2022 Task 1 baseline CNN.

(b) A filter pruning method is applied to compress the "low-complexity CNN" of (a) further. Subsequently, we quantize each parameter from float32 to INT8 data type, reducing networks memory by four times.

(c) An ensemble approach is proposed which combines predictions obtained from several (b) low-complexity CNNs.

(d) Experimental evaluation is undertaken to compare performance of proposed framework with classical methods such as Gaussian Mixture Model (GMMs) and random forest (RF) classifiers and learning based methods such as dictionary learning and a pre-trained high complexity CNN, VGGish with the proposed framework.

The rest of the paper is organized as follows. In Section 2, a brief overview of dataset used and features used for experimentation is described. In Section 3, a procedure to obtain low-complexity CNN and an ensemble framework is described. Section 4 presents experimental analysis. Finally, discussion and conclusion is presented in Section 5 and 6 respectively.

2. EXPERIMENTAL DATASET AND FEATURE EXTRACTION

DCASE 2022 Task 1 uses the TAU Urban Acoustic Scenes 2022 Mobile, development and evaluation datasets [13]. The dataset contains recordings from 12 European cities in 10 different acoustic scenes using 4 different recording devices. Each audio recording has 1 second length. The development dataset is divided in training and validation sets. The training dataset consists of 139620 audio examples and the validation dataset consists of 29680 audio examples. The evaluation dataset consists of only audio examples without any groundtruth available publicly, and the evaluation is performed by the DCASE challenge community.

Feature extraction: For time-frequency representations, log-mel band energies of size (40 × 51) corresponding to an audio signal of 1 second length are extracted. A Hamming asymmetric window of length 40ms, and a hop length of 20ms is used to extract magnitude spectrogram. Next, log-mel spectrogram is computed using 40 mel bands.

3. OBTAINING LOW-COMPLEXITY CNNS

Low-complexity optimal CNN architecture: We design a simple low-complexity CNN architecture which consists of three convolutional layers (C1, C2 & C3), two pooling layers (P1, P2), a dense layer (D) and a classification layer. We perform an empirical analysis to select an appropriate size of CNN filters and activation functions across the different layers. The filter size for each convolutional layer is chosen from a set, $\{1 \times 1, 3 \times 3, 5 \times 5, 7 \times 7\}$. We choose hyperbolic tangent (tanh) or rectified linear unit (ReLU) activation function for different layers. The low-complexity CNN is trained using the training dataset with a batch size of 64 with an Adam optimizer for 1000 epochs. A categorical cross-entropy loss function is used during the training process. We apply an early stopping criterion to yield the best network that gives the minimum log-loss for the validation dataset.

The optimal architecture obtained after performing empirical analysis is given in Table 1. The details of the experiments are given in Section 4. The proposed architecture requires approximately 5M MACs to produce an output corresponding to an input of size (40 x 51), and has 14886 parameters. The performance of the trained network is measured in terms of accuracy and log-loss, averaged over 5 different iterations.

Reducing redundancy in the low-complexity optimal CNN via filter pruning: To eliminate redundant filters from the low-complexity optimal architecture as given in Table 1, we apply a filter pruning strategy. For each convolutional layer, we identify filter pairs which are similar. Our hypothesis is that similar filters produce similar output or feature maps and hence, contribute to redundancy only. Therefore, one of the similar filters can be eliminated. The similarity between the filters is measured using a cosine distance. We identify the closest filter pairs for each layer separately. A filter from each pair is deemed redundant and eliminated from the network. More information about the similarity based filter pruning method can be found at [14].

The number of redundant filters obtained after performing similarity-based filter pruning for C1 layer is 4 out of 16, C2 layer is 4 out of 16 and C3 layer is 10 out of 32. We obtain 6 different pruned networks that are obtained after pruning C1 layer only, C2 layer only, C3 layer only, C1 and C2 layers, C2 and C3 layers, C1 and C2 and C3 layers. The number of MACs and the number of parameters for each pruned network are given in Table 2.

To regain the loss in performance due to pruning, the pruned networks are fine-tuned in a training similar to that of the unpruned low-complexity optimal network.

Reducing memory requirement via quantization: To reduce size of the network, we perform quantization on parameters of each pruned network using Tensorflow-Lite (TFLite). TFlite [15] is an open-source framework to quantizes a pre-trained full-precision network for embedded devices. We quantize the network parameters from 32-bit floating point to 8-bit integers. This leads to reduce the network size 4x.

Table 1: Low-complexity optimal CNN architecture. Here, tanh is a hyperbolic tangent activation function and ReLU is a rectified linear unit activation function.

Layer name	Description
C1	Convolution 16@(3 × 3) + Batch Normalization + tanh
C2	Convolution 16@(3 × 3) + Batch Normalization + ReLU
P1	Average Pooling (5 × 5)
C3	Convolution 32@(3 × 3) + Batch Normalization + tanh
P2	Average Pooling (4 × 10)
D	Dense (100) + tanh
Classification	softmax (10)

An ensemble framework: Next, the predictions obtained from various pruned networks are aggregated together in an ensemble framework. The total number of parameters in the ensemble framework that aggregates predictions from all 6 pruned networks are 70.97K and the total number of MACs are 23.84M.

The codes and the pruned networks can be found online [1][2].

4. PERFORMANCE ANALYSIS

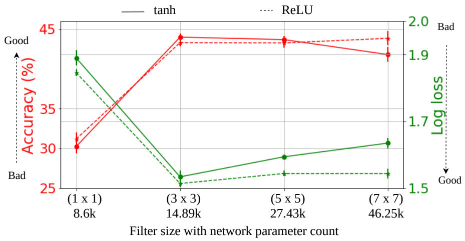


Figure 1: Accuracy and log loss obtained for DCASE 2022 Task 1 development validation dataset using the low-complexity CNNs at different filter size and activation function.

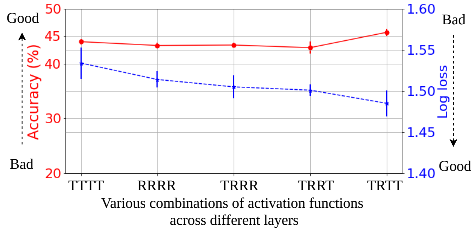


Figure 2: Accuracy and log loss obtained for DCASE 2022 Task 1 development validation dataset using the low-complexity CNN with (3 x 3) filter size when different combinations of activation functions are applied across “C1”, “C2”, “C3” and “D” layers. Here, “T” represents tanh activation function and “R” represents ReLU activation function.

The accuracy and the log-loss obtained at various filter size and different activation functions in the low-complexity CNN is shown in Figure 1. The low-complexity CNN with (3 x 3) filter size outperforms networks with other filter size.

Using “ReLU” activation function across all layers results in smaller log-loss in comparison to that of the the “tanh” activation

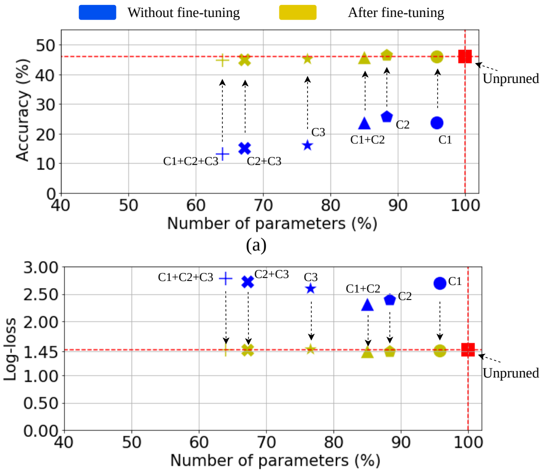
[1] Link: Pruned Quantized models, confusion matrices, evaluation scripts.

[2] Link: Model size and complexity calculation.

Table 2: Various low-complexity CNNs obtained after pruning and applying quantization (INT8).

Sr No.	Network Name	Pruned layer	Architecture (C1-C2-C3-Dense)	Parameters	Size (KB)	MACs (millions)
1	Unpruned optimal low-complexity	NA	16-16-32-100	14886	18.59	5.41
2	Pruned_C1	C1	12-16-32-100	14254	17.86	4.16
3	Pruned_C2	C2	16-12-32-100	13138	16.85	4.13
4	Pruned_C3	C3	16-16-22-100	11396	15.11	5.29
5	Pruned_C12	C1 + C2	12-12-32-100	12650	16.26	3.18
6	Pruned_C23	C2 + C3	16-12-22-100	10008	13.73	4.04
7	Pruned_C123	C1 + C2 + C3	12-12-22-100	9520	13.14	3.08

function. On the other hand, combining "tanh" and "ReLU" activation functions across different layers improves the log-loss further as shown in Figure 2. We find that the "tanh" activation function in the first layer and "ReLU" for other layers improves the performance over that of using "ReLU" for all layers. Finally, we choose the low-complexity optimal CNN architecture which performs better than that of architectures with different combinations of activation function. The optimal low-complexity CNN has "tanh" activation function in all layers except C2 and the filter size is (3 x 3) in each convolutional layers.



(a)

(b)

Figure 3: (a) Accuracy and (b) log-loss obtained after pruning different intermediate layers (C1, C2, C3, C1+C2, C2+C3, C1+C2+C3) in the unpruned low-complexity optimal CNN for DCASE 2022 Task 1 development validation dataset. The accuracy and the log-loss is obtained with and without performing the fine-tuning of the pruned network.

Next, we analyse the performance obtained using the low-complexity optimal CNN without performing any pruning and after pruning it. Figure 3 shows the accuracy and the log-loss obtained for the unpruned low-complexity optimal network and its various pruned networks. The unpruned low-complexity optimal network gives 1.475 log-loss and 45.92% accuracy. Eliminating filters from the unpruned network results in a significant reduction in performance, but this is almost entirely restored after fine-tuning.

The performance obtained after aggregating predictions from various pruned networks is given in Table 3. The ensemble framework improves the performance in comparison to that of individual pruned networks.

Performance comparisons: The proposed ensemble framework improves performance as compared to the DCASE 2022 Task 1 baseline network for development and evaluation datasets as given in Table 4. We also compare the performance with the following methods,

- **GMM:** We train Gaussian Mixture Models (GMMs) for each scene class separately, and perform classification using maximum likelihood estimates. The log-mel spectrogram of size (40 x 51) is averaged along temporal dimension to yield (40 x 1) vector, which is given as an input to train the GMMs. The number of mixtures (n) per class are chosen from $\{5,10,15,20,30,50\}$.
- **RF:** A random forest (RF) classifier is trained using the log-mel spectrogram averaged along the temporal dimension. The number of estimators are set to 100 and the depth (d) is chosen from $\{5,20,40,50,100,200\}$.
- **Dictionary learning**: A dictionary learning framework with structured incoherence and shared features (DLSI) [16] is trained. DLSI framework learns the dictionary for each class by minimizing the reconstruction error and reduces the redundant dictionary atoms in the learning process itself. The number of dictionary atoms per class (k) are chosen from $\{5,10,15,20,40\}$.
- **MLP**: A multi-layer perceptron (MLP) network with 2 dense layers each having 100 and 50 units is trained on averaged log-mel spectrogram along the temporal dimension.
- **Pre-trained CNN**: We use pre-trained convolutional layers of VGGish [17, 18] followed by a dense and a classification layer to yield an end-to-end VGGish network. The input to the end-to-end VGGish is log-mel spectrogram of size (40 x 51). We train the end-to-end VGGish for 1000 epochs with similar training settings as used to train the low-complexity CNNs.

The proposed ensemble framework outperforms the other methods as shown in Figure 4. It is interesting to note that the proposed low-complexity optimal CNN performs better than that of the large-scale pre-trained CNN which has 4.5M parameters and has 1077M MACs.

Similarity analysis among various pruned models: We analyse similarity between the different pruned models as given in Table 2. For this, we use 10k audio examples from the validation dataset, and generated outputs from the various filters (feature maps). The similarity is measured using the aggregated mean square error (MSE) across 10k examples, computed between the corresponding feature maps of two different pruned models.

We find that a few of the feature maps generated by the two different pruned models are similar with MSE $\leq 10^{-4}$. To show

Table 3: Performance obtained for DCASE 2022 Task 1 development validation dataset using the ensemble framework that combines various low-complexity optimal CNNs obtained after pruning and quantization.

Sr No.	Ensemble framework	Number of parameters	MACs (millions)	Size (KB)	Accuracy (%)	Log-loss
1	All pruned networks except Pruned_C1	56712	19.72	75.09	47.14	1.394
2	All pruned networks except Pruned_C2	57828	19.75	76.10	47.10	1.396
3	All pruned networks except Pruned_C3	59570	18.60	77.84	47.26	1.392
4	All pruned networks except Pruned_C12	58316	20.70	76.69	47.45	1.394
5	All pruned networks except Pruned_C23	60958	19.84	79.22	47.52	1.389
6	All pruned networks except Pruned_C123	61446	20.80	79.81	47.35	1.392
7	Ensemble on all pruned networks	70966	23.84	92.95	47.45	1.389

Table 4: Performance comparison

Framework	Dataset			
	Development		Evaluation	
	Accuracy (%)	log-loss	Accuracy (%)	log-loss
DCASE baseline network [12]	42.9	1.575	44.2	1.532
Proposed ensemble (#3 in Table 3)	47.26	1.392	45.9	1.492

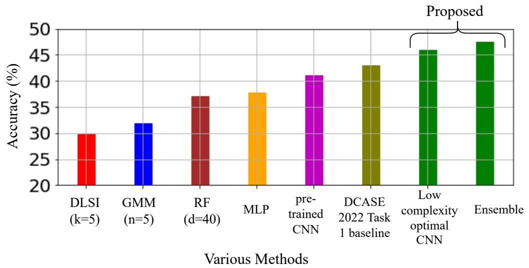


Figure 4: Accuracy comparison of various methods for DCASE 2022 Task 1 development validation set. Here, we show the best accuracy obtained from the DLSI, GMM and RF framework.

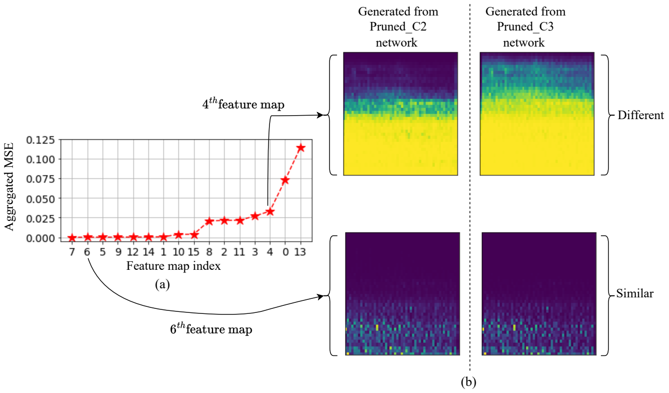


Figure 5: (a) Aggregated MSE computed between the feature maps generated using Pruned_C2 and Pruned_C3 network from the C1 layer. (b) shows the different and the similar feature maps corresponding to 4th and 6th feature map index.

this, the aggregated MSE computed across corresponding feature maps generated by the Pruned_C2 and the Pruned_C3 network is plotted in Figure 5(a). A few of the feature maps have MSE close to zero, and hence these feature maps are similar. We also show the different and the similar feature maps obtained from the Pruned_C2 network and the Pruned_C3 network corresponding to 4th and 6th index for a given input in Figure 5(b). This suggests that similar feature maps across different pruned networks are redundant, and are not required to compute again for the other network.

We find that the Pruned_C1 network shares 7 feature maps with the Pruned_C12 network, and 3 feature maps with the Pruned_C123 network across C1 layer. Similarly, the Pruned_C2 network shares 7 feature maps with the Pruned_C3, network and 6 feature maps with the Pruned_C23.

Ignoring similar feature maps across different pruned models, the MACs could be further reduced by 6M points, and the number of total parameters are reduced by approximately 3.4k in the ensemble framework that combines all pruned networks.

5. DISCUSSION

We observe that small size network obtained after pruning a relatively large size network gives better accuracy compared to that of the similar size network obtained from scratch. For example, the Pruned_C123 network having 9.6k parameters and (3 x 3) filter size gives approximately 16 percentage points more accuracy compared to that of the similar size low-complexity CNN having (1 x 1) filter size and 8.6k parameters as shown in Figure 1.

We find that the ensemble on various CNNs improves the performance as compared to the individual CNN. However, the ensemble on the various CNNs consumes more resources. Therefore, pruning individual CNNs provide an advantage to use the ensemble framework efficiently. To improve the efficiency of the ensemble further, the shared feature maps (similarity) across various CNNs can be ignored.

6. CONCLUSION

This paper focuses on designing a low-complexity system for acoustic scene classification. A filter pruning and quantization is applied to obtain compressed, accelerated, and low-size CNN. Further, various low-size CNNs are combined in the ensemble framework to improve classification performance. The proposed framework shows promising results in terms of reduction in parameters and improved performance. In future, our aim is to improve the performance of the low-complexity framework further.

7. ACKNOWLEDGEMENTS

This work was partly supported by Engineering and Physical Sciences Research Council (EPSRC) Grant EP/T019751/1 "AI for Sound (AI4S)". For the purpose of open access, the authors have applied a Creative Commons Attribution (CC BY) licence to any Author Accepted Manuscript version arising. Thanks to James A. King and Xubo Liu, PhD students at University of Surrey, UK for the discussions.

8. REFERENCES

[1] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, *et al.*, "Recent advances in convolutional neural networks," *Pattern Recognition*, vol. 77, pp. 354–377, 2018.

[2] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *Proceedings of International Conference on Learning Representations, ICLR (arXiv preprint arXiv:1409.1556)*, 2015.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, pp. 1097–1105, 2012.

[4] Y. Wen and D. Gregg, "Exploiting weight redundancy in CNNs: Beyond pruning and quantization," *arXiv preprint arXiv:2006.11967*, 2020.

[5] A. Singh, P. Rajan, and A. Bhavsar, "SVD-based redundancy removal in 1-D CNNs for acoustic scene classification," *Pattern Recognition Letters*, vol. 131, pp. 383–389, 2020.

[6] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf, "Pruning filters for efficient ConvNets," *Proceedings of International Conference on Learning Representations, ICLR*, 2017.

[7] Z. Liu, J. Li, Z. Shen, G. Huang, S. Yan, and C. Zhang, "Learning efficient convolutional networks through network slimming," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2736–2744, 2017.

[8] Y. He, X. Zhang, and J. Sun, "Channel pruning for accelerating very deep neural networks," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1389–1397, 2017.

[9] Y. He, P. Liu, Z. Wang, Z. Hu, and Y. Yang, "Filter pruning via geometric median for deep convolutional neural networks acceleration," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4340–4349, 2019.

[10] M. Lin, R. Ji, Y. Wang, Y. Zhang, B. Zhang, Y. Tian, and L. Shao, "HRank: Filter pruning using high-rank feature map," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1529–1538, 2020.

[11] B. Kim, S. Yang, J. Kim, and S. Chang, "QTI submission to DCASE 2021: Residual normalization for device imbalanced acoustic scene classification with efficient design," *DCASE2021 Challenge, Tech. Rep*, 2021.

[12] I. Martín-Morató, F. Paissan, A. Ancilotto, T. Heittola, A. Mesaros, E. Farella, A. Brutti, and T. Virtanen, "Low-complexity acoustic scene classification in DCASE 2022 Challenge," *arXiv preprint arXiv:2206.03835*, 2022.

[13] T. Heittola, A. Mesaros, and T. Virtanen, "Acoustic scene classification in DCASE 2020 Challenge: Generalization across devices and low complexity solutions," *Proceedings of the Detection and Classification of Acoustic Scenes and Events (DCASE) Workshop*, 2020.

[14] A. Singh and M. D. Plumbley, "A passive similarity based CNN filter pruning for efficient acoustic scene classification," *Interspeech (arXiv preprint arXiv:2203.15751)*, 2022.

[15] R. David, J. Duke, A. Jain, V. Janapa Reddi, N. Jeffries, J. Li, N. Kreeger, I. Nappier, M. Natraj, T. Wang, *et al.*, "TensorFlow Lite Micro: Embedded machine learning for TinyML systems," *Proceedings of Machine Learning and Systems*, vol. 3, pp. 800–811, 2021.

[16] I. Ramirez, P. Sprechmann, and G. Sapiro, "Classification and clustering via dictionary learning with structured incoherence and shared features," *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 3501–3508, 2010.

[17] J. Cramer, H.-H. Wu, J. Salamon, and J. P. Bello, "Look, listen, and learn more: Design choices for deep audio embeddings," *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3852–3856, 2019.

[18] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. Weiss, and K. Wilson, "CNN architectures for large-scale audio classification," *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 131–135, 2017.