URBAN SOUND TAGGING USING CONVOLUTIONAL NEURAL NETWORKS

Technical Report

Sainath Adapa

FindHotel
Amsterdam, Netherlands
adapasainath@gmail.com

ABSTRACT

This technical report outlines our solution to Task 5 of the DCASE 2019 challenge, titled *Urban Sound Tagging*. The objective of the task is to label different sources of noise from raw audio data. A modified form of MobileNetv2, a convolutional neural network (CNN) model was trained to label both coarse and fine tags jointly. The proposed model uses log-scaled Mel-spectrogram as the representation format for the audio data. Mixup, Random erasing, scaling, and shifting are used as data augmentation techniques. A second model that uses scaled labels was built to account for human errors in the annotations. The solution code is available on GitHub[1].

Index Terms— sound event detection (SED), machine listening, audio tagging, convolutional neural networks

1. INTRODUCTION

The Detection and Classification of Acoustic Scenes and Events (DCASE) [1], now in its fifth edition, is a recurring set of challenges aimed at developing computational scene and event analysis methods. In Task 5, Urban Sound Tagging, the objective is to predict the presence or absence of 23 different types of noise sources in audio recordings. The 23 *fine-grained* tags are further grouped into a list of 7 *coarse-grained* tags. This hierarchical relationship is illustrated in Figure 1.

For this challenge, SONYC [2] has provided 2351 recordings as part of the *train* set, and 443 recordings as a part of the *validate* set. All the recordings are ten seconds in length. Each recording was annotated by three Zooniverse[2] volunteers. Additional annotations, specifically for *validate* set, were performed by the SONYC team members and ground truth is then agreed upon by the SONYC team.

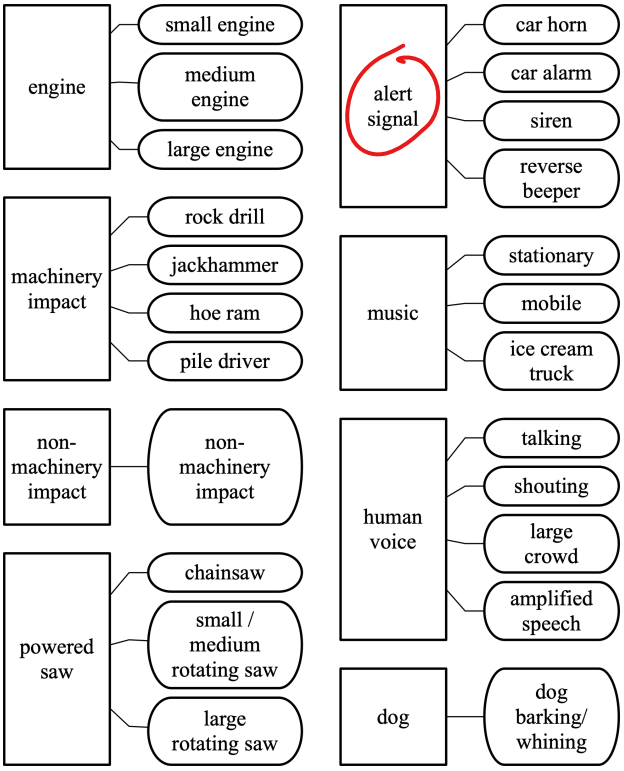

Figure 1: Hierarchical taxonomy of tags. Rectangular and round boxes respectively denote coarse and fine tags. [3]

2. PROPOSED FRAMEWORK

2.1. CNN Architecture

Convolutional neural network (CNN) based architectures have been proven to be useful for audio classification [4, 5]. In this work, we use a modified form of MobileNetV2 [6]. The architecture of MobileNetV2 contains a 2D convolution layer at the beginning, followed by 19 *Bottleneck residual blocks* (shown in Table 1). Spatial average of the output from the final residual block is computed and used for classification via a linear layer.

The proposed model contains few modifications to the above-described architecture. The input Log Mel-spectrogram data is sent to the MobileNetV2 after first passing the input through two convolution layers. This is so that the single channel input can be converted into a three channel input. Instead of the spatial average, Max pooling is applied on the output from the final residual block. Additionally, the single linear layer at the end is replaced by two linear layers. The full architecture is described in Table 2.

All the unmodified layers are initialized with weights from the MobileNetV2 model trained on ImageNet [7]. Kaiming initialization [8] is used for the remaining layers.

[1]https://github.com/sainathadapa/urban-sound-tagging
[2]https://www.zooniverse.org/

Input	Operator	Output
$h \times w \times k$	1x1 ,	$h \times w \times (tk)$
$h \times w \times tk$	3x3 s=s,	$\frac{h}{s} \times \frac{w}{s} \times (tk)$
$\frac{h}{s} \times \frac{w}{s} \times tk$	linear 1x1	$\frac{h}{s} \times \frac{w}{s} \times k'$

Table 1: *Bottleneck residual block* transforming from k to k' channels, with stride s, and expansion factor t.

Operator	t	c	n	s
conv2d	**-**	**10**	**1**	**1**
conv2d	**-**	**3**	**1**	**1**
conv2d	-	32	1	2
bottleneck	1	16	1	1
bottleneck	6	24	2	2
bottleneck	6	32	3	2
bottleneck	6	64	4	2
bottleneck	6	96	3	1
bottleneck	6	160	3	2
bottleneck	6	320	1	1
conv2d 1x1	-	1280	1	1
maxpool	**-**	**1280**	**1**	**-**
linear	**-**	**512**	**1**	**-**
linear	**-**	**k**	**1**	**-**

Table 2: Each line describes a sequence of 1 or more identical (modulo stride) layers, repeated n times. All layers in the same sequence have the same number c of output channels. The first layer of each sequence has a stride s and all others use stride 1. All spatial convolutions use 3×3 kernels (except for the first two which use 1×1 kernels). The expansion factor t is always applied to the input size as described in Table 1. Modifications to the MobileNetV2 architecture are highlighted in bold.

2.2. Preprocessing and Data augmentation

The proposed model uses Log Mel-spectrogram as the representation format for the input data. Librosa [9] toolbox was used to compute the Mel-spectrogram. For the Short-time Fourier transform (STFT), window size of 256 and hop length of 694 was used. For the Mel-frequency bins computation, lowest frequency and highest frequency of 20Hz and 22050Hz was used, with the number of bins being 128[3]. No re-sampling or additional preprocessing steps were performed.

Several data augmentation techniques were used to supplement the training data. Deformations such as Time stretching and Pitch shifting that were previously shown to help in sound classification were employed [5]. In addition, image augmentation methods such as Random rotate, Grid distortion [10], and Random erasing [11] were used. Mixup [12], an approach that linearly mixes two random training examples was used as well.

3. RE-LABELING

For the *validate set*, we have access to both the ground truth and the three annotations by Zooniverse volunteers. When the ground truth of a label is positive, 36% of annotations (by Zooniverse volunteers) do not match with the ground truth. If the quality of the labels can be improved, it is quite possible that the accuracy of the model can be improved as well. Hence, a logistic regression model that takes the annotations as input and estimates the ground truth label was developed. This model was trained on the *validate* set and then the ground truth estimate for the *train set* were generated. Table 3 shows some predictions from the model.

[3]https://www.kaggle.com/daisukelab/fat2019_prep_mels1

Coarse label	Fine label	Positive annotations count	Predicted score
music	uncertain	1	0.10
music	uncertain	3	0.98
music	stationary	2	0.88
powered saw	chainsaw	3	0.98
machinery impact	-	0	0.05

Table 3: Predictions for few cases from the automatic re-labeling model

4. TRAINING

Two models were trained for this challenge:

- The first model generates probabilities for both the fine and coarse labels. During training, whenever the annotation is "unknown/other", loss for the fine tags corresponding to this coarse tag was masked out. Hence, this model does not generate predictions for the *uncertain* fine labels. For each training example, loss is computed against each set of annotation separately. Average of the three loss values is taken as the loss value for this training example.
- For the second model, predictions from the re-labeling model described in Section 3 are used as labels. This model generates probabilities for both the fine and coarse labels, including the *uncertain* fine labels.

Both the models use identical input data representation, and employ the same data augmentation techniques. They also use Binary Cross-entropy loss as the optimization metric. The models are trained on the *train set* using the *validate set* to determine the stopping point.

Training was done on PyTorch [13]. AMSGrad variant of the Adam algorithm [14, 15] with a learning rate of 1e-3 was utilized for optimization. Whenever the loss on *validate* set stopped improving for 5 *epochs*, learning rate was reduced by a factor of 10. At the time of prediction, test-time augmentation (TTA) in the form of Time shifting was used.

5. RESULTS

The baseline system in [3] computes VGGish embeddings [4] of the audio files, and builds a multi-label logistic regression on top of the embeddings. An additional baseline system that trains a CNN on the log Mel-spectrogram was described in [16]. Both the baseline systems count a positive for a tag if at least one annotator has labeled the audio clip with that tag. Table 4 shows the performance of the two baseline systems compared against the proposed models. It can be observed that re-labeling helped improve the Micro-AUPRC and the Micro-F1 metrics in case of Fine-grained labels.

	FINE-GRAINED			COARSE-GRAINED		
	Micro AUPRC	Micro F1	Macro AUPRC	Micro AUPRC	Micro F1	Macro AUPRC
Baseline - 1 (VGGish [3])	0.672	0.502	0.427	0.762	0.674	0.542
Baseline - 2 (CNN9-avg [16])	0.672	0.371	0.433	0.782	0.519	0.628
modified MobileNetV2 (no re-labeling)	0.772	0.489	0.594	0.861	0.602	0.702
modified MobileNetV2 (with re-labeling)	0.784	0.636	0.570	0.860	0.740	0.700

Table 4: Performance on *validate* set

6. REFERENCES

[1] http://dcase.community/challenge2019/.

[2] J. P. Bello, C. Silva, O. Nov, R. L. DuBois, A. Arora, J. Salamon, C. Mydlarz, and H. Doraiswamy, "Sonyc: A system for the monitoring, analysis and mitigation of urban noise pollution," *arXiv preprint arXiv:1805.00889*, 2018.

[3] http://dcase.community/challenge2019/task-urban-sound-tagging.

[4] S. Hershey, S. Chaudhuri, D. P. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, *et al.*, "Cnn architectures for large-scale audio classification," in *2017 ieee international conference on acoustics, speech and signal processing (icassp)*. IEEE, 2017, pp. 131–135.

[5] J. Salamon and J. P. Bello, "Deep convolutional neural networks and data augmentation for environmental sound classification," *IEEE Signal Processing Letters*, vol. 24, no. 3, pp. 279–283, 2017.

[6] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520.

[7] https://github.com/tensorflow/models/blob/master/research/slim/nets/mobilenet/README.md.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[9] B. McFee, M. McVicar, S. Balke, V. Lostanlen, C. Thom, C. Raffel, D. Lee, K. Lee, O. Nieto, F. Zalkow, D. Ellis, E. Battenberg, R. Yamamoto, J. Moore, Z. Wei, R. Bittner, K. Choi, nullmightybofo, P. Friesch, F.-R. Stter, Thassilo, M. Vollrath, S. K. Golu, nehz, S. Waloschek, Seth, R. Naktinis, D. Repetto, C. F. Hawthorne, and C. Carr, "librosa/librosa: 0.6.3," Feb. 2019. [Online]. Available: https://doi.org/10.5281/zenodo.2564164

[10] E. K. V. I. I. A. Buslaev, A. Parinov and A. A. Kalinin, "Albumentations: fast and flexible image augmentations," *ArXiv e-prints*, 2018.

[11] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation," *arXiv preprint arXiv:1708.04896*, 2017.

[12] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.

[13] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," in *NIPS Autodiff Workshop*, 2017.

[14] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[15] S. J. Reddi, S. Kale, and S. Kumar, "On the convergence of adam and beyond," *arXiv preprint arXiv:1904.09237*, 2019.

[16] Q. Kong, Y. Cao, T. Iqbal, Y. Xu, W. Wang, and M. D. Plumbley, "Cross-task learning for audio tagging, sound event detection and spatial localization: Dcase 2019 baseline systems," *arXiv preprint arXiv:1904.03476*, 2019.