# URBAN SOUND TAGGING USING CONVOLUTIONAL NEURAL NETWORKS

*Sainath Adapa*

FindHotel
Amsterdam, Netherlands
adapasainath@gmail.com

## ABSTRACT

In this paper, we propose a framework for environmental sound classification in a low-data context (less than 100 labeled examples per class). We show that using pre-trained image classification models along with the usage of data augmentation techniques results in higher performance over alternative approaches. We applied this system to the task of Urban Sound Tagging, part of the DCASE 2019. The objective was to label different sources of noise from raw audio data. A modified form of MobileNetV2, a convolutional neural network (CNN) model was trained to classify 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 proposed model achieved the first rank on the leaderboard with Micro-AUPRC values of 0.751 and 0.860 on fine and coarse tags, respectively.

***Index Terms*—** DCASE, machine listening, audio tagging, convolutional neural networks

## 1. INTRODUCTION

The IEEE AASP challenge on 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 tags in audio recordings. Each of these tags represents a source of noise and thus a cause of noise complaints in New York City. Solutions for this task, such as the one proposed in this paper, will help inspire the development of solutions for monitoring, analysis, and mitigation of urban noise.

## 2. RELATED WORK

The current task of Urban Sound Tagging is part of the broader research area of Environmental Sound Classification [1]. Convolutional neural networks (CNNs) that use Log-scaled Mel-spectrogram as the feature representation have been proven to be useful for this use case [2, 3], and have also achieved leading performance in recent DCASE tasks [4, 5, 6]. Extensions to the CNN framework, in the form of Convolutional Recurrent Neural Networks (CRNNs) have been proposed [7]. Transformation of the raw audio waveform into the Mel-spectrogram representation is a ”lossy” operation [8]. As such, there has been ongoing research into evaluating alternatives such as using Scattering transform [9], Gammatone filter bank [7] representations, as well as directly employing one-dimensional CNN on the raw audio signal [10]. Operating in the context of noisy labels [11] or in a low-data regime [12] (both of which are properties of the present task) are two other active research areas in this domain. One particular approach for dealing with small labeled datasets is the usage of pre-trained models to generate embeddings that can be used for downstream audio classification tasks. VGGish[13], SoundNet[14], and $L^3$-Net[15] are examples of such models.

## 3. DATASET

For this challenge, SONYC [16] has provided 2351 recordings as part of the *train set*, and 443 recordings as a part of the *validate set*. All the recordings, acquired from different acoustic sensors in New York City, are Mono channel, sampled at 44.1kHz, and are ten seconds in length. The private *evaluation set* consisted of 274 recordings. Labels for these recordings were revealed only at the end of the challenge. A single recording might contain multiple noise sources. Hence, this is a task of multi-label classification.

The 23 noise tags, termed *fine-grained tags*, are further grouped into a list of 7 *coarse-grained tags*. This hierarchical relationship is illustrated in Figure 1. 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. Since the *fine-grained tags* are not always easily distinguishable, annotators were given the choice of assigning seven tags of the form ”other/unknown” for such cases. Each of these seven tags termed ”incomplete tags,” correspond to a different coarse category.

## 4. PROPOSED FRAMEWORK

### 4.1. CNN Architecture

In this work, we use a modified form of MobileNetV2 [18]. The architecture of MobileNetV2 contains a 2D convolution layer at the beginning, followed by 19 *Bottleneck residual blocks* (described 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 makes few modifications to the above-described architecture. The input Log Mel-spectrogram data is sent to the MobileNetV2 after passing it through two convolution layers. This process transforms the single-channel input into a three-channel tensor to match the input size of original MobileNetV2 architecture. Instead of the spatial average, Max pooling is applied

[1] http://dcase.community/

[2] https://www.zooniverse.org/

https://doi.org/10.33682/8axe-9243

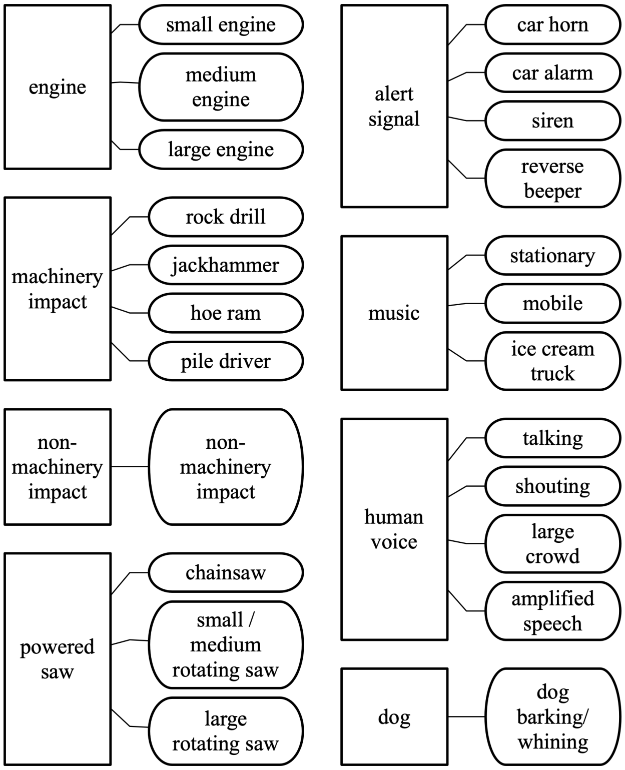

Figure 1: Hierarchical taxonomy of tags. Rectangular and round boxes respectively denote coarse and fine tags respectively. [17]

| 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 $t$.

to 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.

### 4.2. Initialization with Pre-trained weights

In many fields, including in the acoustic area, CNNs exhibit better performance with an increase in the number of layers [19, 20]. However, it has been observed that deeper neural networks are harder to train and prone to overfitting, especially in the context of limited data [21].

Many of the *fine-grained tags* have less than 100 training examples with positive annotations, thus placing the current task into a *low-data regime* context [12]. Since the proposed architecture has a large (24) number of layers, we initialized all the unmodified layers of the network with weights from the MobileNetV2 model trained on ImageNet [22, 23]. Kaiming initialization [24] is used for the remaining layers. Since the domain of audio classification is different from image classification, we do not employ a Fine-tuning approach [25] here. All the layers are jointly trained from the beginning. When all the layers with ImageNet weights were frozen at that parameters, the model performed worse than the baseline model (Section 6) showing the need for joint training of the whole network.

| 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 $3 \times 3$ kernels (except for the first two which use $1 \times 1$ 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.

| | ImageNet pre-trained weights | Kaimin initialization |
|---|---|---|
| *train set* loss | 0.1401 ± 0.0017 | 0.1493 ± 0.0019 |
| *validate set* loss | 0.1200 ± 0.0008 | 0.1266 ± 0.0022 |

Table 3: Final Binary Cross-entropy loss values at the end of training. 5 repetitions of training runs from scratch were performed.

The rationale behind the use of ImageNet weights is that the kind of filters that the ImageNet based model has learned are applicable in the current scenario of Spectrograms as well. Especially the filters in the initial layers that detect general patterns like edges and textures[26] are easily transferable to the present case. With the described initialization, we noticed faster and better convergence (illustrated in Figure 2 and Table 3) when compared to initializing all the layers with Kaimin initialization. Similar gains were observed previously in the context of Acoustic Bird Detection [3].

Other pre-trained models such as ResNeXt[27], and EfficientNet[28] were also tested. The observed metrics were at the same level as the MobileNetV2 architecture. Since the performance is similar, MobileNetV2 was chosen as it has the least number of parameters among the models tried.

### 4.3. Preprocessing and Data augmentation

The proposed model uses Log Mel-spectrogram as the representation format for the input data. Librosa [29] toolbox was used to compute the Mel-spectrogram. For the Short-time Fourier transform (STFT), *window length* of 2560 and *hop length* of 694 was

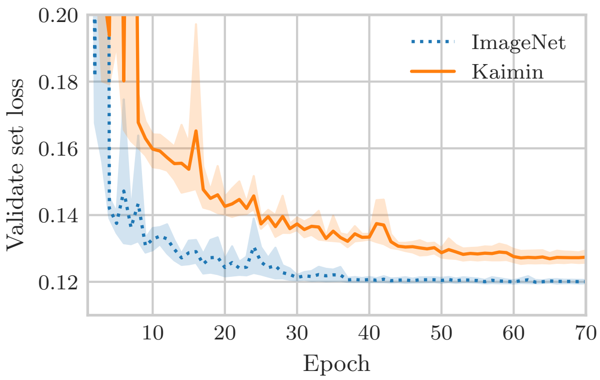

Figure 2: Trajectory of *validate set* loss during training, demonstrating that using pre-trained ImageNet weights results in faster convergence.

| | Fine-level Micro-AUPRC | Coarse-level Micro-AUPRC |
|---|---|---|
| No data augmentation | 0.716 | 0.819 |
| Only Mixup | 0.745 | 0.840 |
| Only Random erasing | 0.732 | 0.820 |
| Only Random rotate | 0.728 | 0.832 |
| Only Shifting time | 0.719 | 0.822 |
| Only Grid distortion | 0.753 | 0.842 |
| Pitch shifting and Time stretching | 0.732 | 0.834 |
| All the techniques | 0.772 | 0.855 |

Table 4: Performance on the *validate set*, demonstrating the gains due to data augmentation

used. For the Mel-frequency bins computation, the lowest and the highest frequencies were set at 20Hz and 22050Hz, respectively, 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 [2]. Also, image augmentation methods such as Random rotate, Grid distortion [30], and Random erasing [31] were used. Mixup [32], an approach that linearly mixes two random training examples was used as well. Table 4 shows the impact of Data augmentations, when each of the methods were applied separately.

### 4.4. Re-labeling

For the *validate set*, we have access to both the ground truth and the three sets of 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 increased 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* was generated. Table 5 shows a sample of predictions from the model.

| 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 5: Predictions for few cases from the automatic re-labeling model

## 5. MODEL TRAINING

### 5.1. Evaluation metric

Area under the precision-recall curve using the micro-averaged precision and recall values (Micro-AUPRC) is used as the classification metric for this task. Micro-F1 and Macro-AUPRC values are reported as secondary metrics. Detailed information about the evaluation process is available on the task website [17].

### 5.2. Training

Two models were trained for this challenge:

**M1:** 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 *uncertain* fine labels. Since there are three sets of annotations for each training example, one by each Zooniverse volunteer, the loss is computed against each annotation set separately. Average of the three loss values is taken as the final loss value for a training example.

**M2:** For the second model, predictions from the re-labeling model described in Section 4.4 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 (mentioned in Section 4.3). 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 [33]. AMSGrad variant of the Adam algorithm [34, 35] with a learning rate of 1e-3 was utilized for optimization. Whenever the loss on *validate* set stopped improving for five *epochs*, the learning rate was reduced by a factor of 10. Regularization in the form of Early stopping was used to prevent overfitting [36]. At the time of prediction, test-time augmentation (TTA) in the form of Time shifting was used.

## 6. RESULTS

The baseline system mentioned on the task page [17] computes VGGish embeddings [13] of the audio files and builds a multi-label

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

| | FINE-LEVEL PREDICTION | | | COARSE-LEVEL PREDICTION | | |
|---|---|---|---|---|---|---|
| | Macro AUPRC | Micro F1 | Micro AUPRC | Macro AUPRC | Micro F1 | Micro AUPRC |
| Baseline | 0.531 | 0.450 | 0.619 | 0.619 | 0.664 | 0.742 |
| M1 | 0.645 | 0.484 | 0.751 | 0.718 | 0.631 | 0.860 |
| M2 | 0.622 | 0.575 | 0.721 | 0.723 | 0.745 | 0.847 |

Table 6: Performance on the private *evaluation set*

| | COARSE-LEVEL PREDICTION | | | FINE-LEVEL PREDICTION | | |
|---|---|---|---|---|---|---|
| | Baseline | M1 | M2 | Baseline | M1 | M2 |
| Engine | 0.832 | 0.888 | 0.878 | 0.638 | 0.665 | 0.673 |
| Machinery impact | 0.454 | 0.627 | 0.578 | 0.539 | 0.718 | 0.604 |
| Non-machinery impact | 0.170 | 0.361 | 0.344 | 0.182 | 0.362 | 0.374 |
| Powered saw | 0.709 | 0.684 | 0.643 | 0.478 | 0.486 | 0.378 |
| Alert signal | 0.727 | 0.897 | 0.875 | 0.543 | 0.858 | 0.832 |
| Music | 0.246 | 0.404 | 0.586 | 0.168 | 0.289 | 0.351 |
| Human voice | 0.886 | 0.947 | 0.949 | 0.777 | 0.841 | 0.833 |
| Dog | 0.929 | 0.937 | 0.931 | 0.922 | 0.936 | 0.931 |

Table 7: Class-wise AUPRC on the private *evaluation set*

logistic regression model on top of the embeddings. For this baseline system, a label for an audio recording is considered positive if at least one annotator has labeled the audio clip with that tag. Table 6 shows the performance of the baseline system compared against the proposed models on the private *evaluation set*. The proposed models[4] exhibit improved Micro-AUPRC values for both *fine-grained* and *coarse-grained* labels when compared against the baseline model. Moreover, it can be observed that re-labeling didn't prove effective; it helped improve the Micro-F1 score significantly, but it didn't help raise Micro-AUPRC or Macro-AUPRC.

Class-wise AUPRC performance is reported in Table 7. The modified MobileNetV2 architecture improves over the Baseline model performance for all classes (except one) at both coarse and fine-level prediction. In the case of coarse-level prediction, the AUPRC performance for "Powered saw" is lesser than that of Baseline.

## 7. CONCLUSIONS AND FUTURE DIRECTIONS

In this paper, we presented our solution to Task 5 (Urban Sound Tagging) of the DCASE 2019 challenge. Our approach involved using a pre-trained image classification model and modifying it for audio classification. We also employed data augmentation techniques to help with the training process. This resulted in our model achieving Micro-AUPRC values of 0.751 and 0.860 on Fine and Coarse tags, respectively thus obtaining the first rank on the leaderboard. We thus demonstrated that impressive gains could be made when compared to using audio embeddings, even in a low-resource scenario such as the one presented here.

As noted in [37], AUPRC only partially correlates with cross-entropy, i.e., decrease in Binary cross-entropy loss may not always result in increase in AUPRC. Exploring loss functions that are more related to AUPRC metric is an avenue for improvement. Depending on the type of class to be predicted, different input representations (such as STFT, HPSS, Log-Mel) might be better [38]. Thus, an ensemble model that uses these different representations can surpass the one proposed in this paper. This ensemble can also involve models that use VGGish or L$^3$-Net embeddings.

## 8. REFERENCES

[1] O. Houix, G. Lemaitre, N. Misdariis, P. Susini, and I. Urdapilleta, "A lexical analysis of environmental sound categories." *Journal of Experimental Psychology: Applied*, vol. 18, no. 1, p. 52, 2012.

[2] 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.

[3] M. Lasseck, "Acoustic bird detection with deep convolutional neural networks," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2018 Workshop (DCASE2018)*, November 2018, pp. 143–147.

[4] I.-Y. Jeong and H. Lim, "Audio tagging system using densely connected convolutional networks," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2018 Workshop (DCASE2018)*, November 2018, pp. 197–201.

[5] O. Akiyama and J. Sato, "Multitask learning and semi-supervised learning with noisy data for audio tagging," DCASE2019 Challenge, Tech. Rep., June 2019.

[6] H. Chen, Z. Liu, Z. Liu, P. Zhang, and Y. Yan, "Integrating the data augmentation scheme with various classifiers for acoustic scene modeling," DCASE2019 Challenge, Tech. Rep., June 2019.

[7] Z. Zhang, S. Xu, T. Qiao, S. Zhang, and S. Cao, "Attention based convolutional recurrent neural network for environmental sound classification," *arXiv preprint arXiv:1907.02230*, 2019.

[8] L. Wyse, "Audio spectrogram representations for processing with convolutional neural networks," *arXiv preprint arXiv:1706.09559*, 2017.

[9] J. Salamon and J. P. Bello, "Feature learning with deep scattering for urban sound analysis," in *2015 23rd European Signal Processing Conference (EUSIPCO)*. IEEE, 2015, pp. 724–728.

[10] S. Abdoli, P. Cardinal, and A. L. Koerich, "End-to-end environmental sound classification using a 1d convolutional neural network," *Expert Systems with Applications*, 2019.

[11] E. Fonseca, M. Plakal, D. P. Ellis, F. Font, X. Favory, and X. Serra, "Learning sound event classifiers from web audio with noisy labels," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 21–25.

[12] J. Pons, J. Serrà, and X. Serra, "Training neural audio classifiers with few data," *ArXiv*, vol. 1810.10274, 2018.

[13] 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.

[4] https://github.com/sainathadapa/urban-sound-tagging

[14] 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.

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

[16] J. P. Bello, C. Silva, O. Nov, R. L. Dubois, A. Arora, J. Salamon, C. Mydlarz, and H. Doraiswamy, "Sonyc: A system for monitoring, analyzing, and mitigating urban noise pollution," *Communications of the ACM*, vol. 62, no. 2, pp. 68–77, Feb 2019.

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

[18] 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.

[19] D. Eigen, J. Rolfe, R. Fergus, and Y. LeCun, "Understanding deep architectures using a recursive convolutional network," *arXiv preprint arXiv:1312.1847*, 2013.

[20] F. Seide, G. Li, and D. Yu, "Conversational speech transcription using context-dependent deep neural networks," in *Twelfth annual conference of the international speech communication association*, 2011.

[21] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[22] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

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

[24] 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.

[25] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[26] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*. Springer, 2014, pp. 818–833.

[27] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1492–1500.

[28] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," *arXiv preprint arXiv:1905.11946*, 2019.

[29] 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

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

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

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

[33] 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.

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

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

[36] L. Prechelt, "Early stopping-but when?" in *Neural Networks: Tricks of the trade*. Springer, 1998, pp. 55–69.

[37] C. Gousseau, "VGG CNN for urban sound tagging," DCASE2019 Challenge, Tech. Rep., September 2019.

[38] J. Bai and C. Chen, "Urban sound tagging with multi-feature fusion system," DCASE2019 Challenge, Tech. Rep., September 2019.