INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

Intra-Utterance Similarity Preserving Knowledge Distillation for Audio Tagging

Chun-Chieh Chang[1,2], *Chieh-Chi Kao*[2], *Ming Sun*[2], *Chao Wang*[2]

[1] Center for Language and Speech Processing, Johns Hopkins University
[2] Alexa Speech, Amazon.com Inc.

`cchunch1@jhu.edu, {chiehchi, mingsun, wngcha}@amazon.com`

Abstract

Knowledge Distillation (KD) is a popular area of research for reducing the size of large models while still maintaining good performance. The outputs of larger teacher models are used to guide the training of smaller student models. Given the repetitive nature of acoustic events, we propose to leverage this information to regulate the KD training for Audio Tagging. This novel KD method, Intra-Utterance Similarity Preserving KD (IUSP), shows promising results for the audio tagging task. It is motivated by the previously published KD method: Similarity Preserving KD (SP). However, instead of preserving the pairwise similarities between inputs within a mini-batch, our method preserves the pairwise similarities between the frames of a single input utterance. Our proposed KD method, IUSP, shows consistent improvements over SP across student models of different sizes on the DCASE 2019 Task 5 dataset for audio tagging. There is a 27.1% to 122.4% percent increase in improvement of micro AUPRC over the baseline relative to SPs improvement of over the baseline.

Index Terms: knowledge distillation, audio tagging, Intra-Utterance Similarity Preserving

1. Introduction

In recent years, the release of commercial products such as Amazon Echo and Google Home has highlighted the need for compact neural network models. These devices offer a wide range of services that involve interacting with the user based on audio. The neural network models need to be small enough to fit on the device and yet accurate enough to be commercially viable. These smart devices have limited CPU and memory so the number of Floating Point Operations (FLOPs) and the number of model parameters are of great concern.

Knowledge Distillation (KD) is one popular technique used to improve results of smaller models. It works by having a large teacher model guide the training of a smaller student model. One of the first KD methods proposed uses logits from teacher and student models as an additional loss function [1]. The output logits of the teacher can be viewed as soft targets for the student to achieve. This type of guided learning is not just limited to comparing outputs. There are also methods that compare the intermediate output features from selected layers within the teacher and student model. FitNet uses the outputs from an intermediate layer of the neural network and computes the Mean Squared Error between the intermediate outputs of the teacher model and the student model [2]. There are numerous other KD methods but idea is that the teacher model guides the student model through an additional loss function based on a comparison between the two models [3] [4] [5] [6] [7]. Many of these KD methods are published using Image Classifications datasets like CIFAR-10 or CIFAR-100 [8].

There have been previous studies of KD for acoustic related tasks such as Speech Recognition [9] [10] [11] [12], Acoustic Event Detection [13] [14] [15], and Acoustic Scene Classification [16] [17]. Our novel method, Intra-Utterance Similarity Preserving (IUSP) KD, attempts to leverage prior knowledge about the acoustic events when performing KD. In the Audio Tagging, there is often a lot of repetition that can be seen in the spectrogram of the audio clips. This is most apparent when viewing the spectrograms of stationary signals such as ‘car alarms’ and ‘sirens’. For example, Figure 1 is the spectrogram of a siren from the DCASE 2019 Task 5 challenge [18].

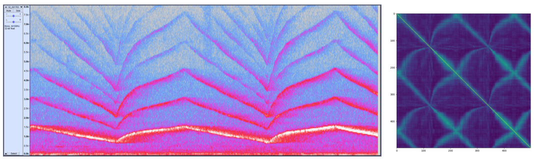

Figure 1: *Left: Example spectrogram of audio clip for siren. Right: Intra-Utterance similarity matrix of siren clip.*

When computing the pairwise similarity between frames, the resultant matrix should have strong values on the off-diagonals. This is seen in the Figure 1. Our proposed method compares the resultant similarity matrices from the intermediate features of both the teacher and student models. This ensures that the student model also captures information about the repetitive nature of the acoustic event.

The rest of the paper will be as follows: Section 2, an explanation of both Similarity Preserving KD (SP) and Intra-Utterance Similarity Preserving KD (IUSP); Section 3, a description of the audio tagging dataset used; Section 4, an overview of the experimental setup; Section 5, a presentation of the results; and Section 6, the conclusion.

2. Method Description

Section 2.1 describes the Similarity Preserving KD (SP) from literature [5] which inspired our Intra-Utterance Similarity Preserving KD (IUSP) described in Section 2.2.

2.1. Similarity Preserving KD

Using the Audio Tagging task as an example, the principle behind Similarity Preserving KD [5] is that inputs with the same event tags should have similar activations in the layers of the neural network. Given an intermediate output $A^{(l)} \in R^{b\times c\times h\times w}$, define $Q^{(l)} \in R^{b\times chw}$ as a reshaping of $A^{(l)}$. Where l is the layer number, b is the batch size, c is the out-

[1] Work done while at Amazon.com Inc.

http://dx.doi.org/10.21437/Interspeech.2020-2835

put channels, and h and w are the input dimensions. In our task, h is related to number of log mel-frequency bins and w is related to the number of frames. Note that various layers may reduce h and w so they may not have the exact same values as the original input dimensions.

The pairwise similarities between each audio clip in the batch can be computed using the following equations:

$$\tilde{G}^{(l)} = Q^{(l)} \cdot Q^{(l)\top};\ G^{(l)}_{[i,:]} = \tilde{G}^{(l)}_{[i,:]} / ||\tilde{G}^{(l)}_{[i,:]}||_2 \qquad (1)$$

Where i denotes the row and so $G^{(l)}$ is the row-wise normalized version of $\tilde{G}^{(l)}$. Since the goal of Similarity Preserving KD is to ensure that the student learns the same pairwise similarity matrix as the teacher, this matrix $G^{(l)}$ is computed for both the teacher and the student models. The Similarity Preserving KD loss is thus defined:

$$\mathcal{L}_{SP} = \frac{1}{b^2} \sum_{(l,l') \in \mathcal{I}} ||G^{(l)}_{Teacher} - G^{(l')}_{Student}||^2_F \qquad (2)$$

Various different layers of the student and teacher model can be compared against each other so $\mathcal{I}$ is the collection of the layer pairs. With l being the layer index of the teacher model and l' being the layer index of the student model.

2.2. Intra-Utterance Similarity Preserving KD

As mentioned in the Introduction, our proposed method is based on Similarity Preserving KD [5]. However, instead of preserving the pairwise similarities between utterances in a batch, we preserve the pairwise similarities between frames of an utterance.

Given an intermediate output $A^{(l)} \in R^{b \times c \times h \times w}$ as defined in Section 2.1 above. Since $A^{(l)}$ includes the entire batch, call the integer $b' \in [1, b]$ the index of a specific utterance in the batch. First, normalize $A^{(l)}$ along the channel dimension using an equation shown below. This way no channel has greater weight than the others. Then, compute the similarity matrix between frames, also shown below.

$$G_{(b')} = Q^{(l)}_{[b']} \cdot Q^{(l)\top}_{[b']};\ \tilde{A}^{(l)}_{[b',:,:,:]} = A^{(l)}_{[b',i,:,:]} / ||A^{(l)}_{[b',i,:,:]}||_2 \quad (3)$$

$Q^{(l)}_{[b']} \in R^{ch \times w}$ is defined as a reshaped version of $\tilde{A}^{(l)}_{[b',:,:,:]} \in R^{c \times h \times w}$. The matrix $G_{(b')}$ is the pairwise similarity between each frame of utterance b' in the batch. This similarity matrix is computed for both the student and teacher model. In the event that h and w from the teacher model are different than those of the student model, bilinear interpolation is used to make the dimensions of $A^{(l)}_{Teacher}$ match $A^{(l)}_{Student}$. The Intra-Utterance Similarity Preserving KD loss function thus is defined:

$$\mathcal{L}_{IUSP} = \frac{1}{b} \sum_{b' \in [1,b]} ||G^{Teacher}_{(b')} - G^{Student}_{(b')}||^2_F \qquad (4)$$

One modification made to the loss function above was the addition of a sigmoid function applied to each element in matrix $G_{(b')}$. This ensures that the differences between high and low similarities are exaggerated. The final loss function is defined as:

$$\mathcal{L}_{IUSP} = \frac{1}{b} \sum_{b' \in [1,b]} ||\tilde{G}^{Teacher}_{(b')} - \tilde{G}^{Student}_{(b')}||^2_F \qquad (5)$$

Where:

$$\tilde{G}_{(b')} = sig(\gamma \times (G_{(b')} - \delta)) \qquad (6)$$

The hyper parameters γ and δ are meant to scale and shift the sigmoid function so that it can serve as a threshold for determining high and low similarity. We selected the values 10 and 0.5 for γ and δ respectively after tuning on the validation data of the dataset we used.

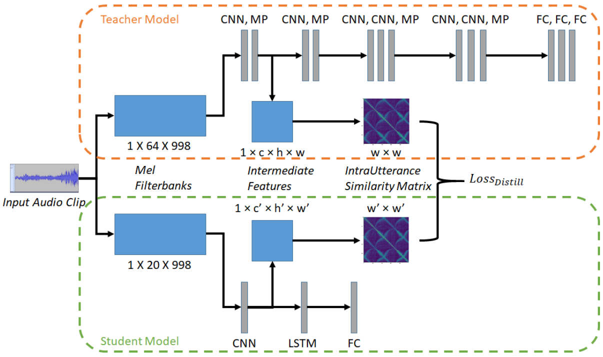

Figure 2: *IntraUtterance Similarity Preserving KD. MP is Max Pooling and FC is Fully Connected Layer. The pairwise similarity of each frame in a given input of a batch is computed for both the teacher and student network. These matrices are then compared against each other. This IntraUtterance Similarity Preserving KD can be used alongside other KD methods.*

3. Dataset Description

The dataset used is the DCASE 2019 Task 5 dataset [18]. It is an Audio Tagging task where the goal is to predict whether or not any of the 8 coarse-grained classes are present in the 10s audio clip. The given development data is split into 2351 clips of training data and 443 clips of validation data. The 274 clips of test data was released after the competition. The metric used for this competition was Area Under Precision and Recall Curve (AUPRC). The higher the AUPRC the better. The detection threshold was incrementally raised and the global tally of true positives (TP), false positives (FP), and false negatives (FN) at that threshold was computed by summing together the individual tallies of TP, FP, and FN of each category for the same threshold. This global TP, FP, and FN at each threshold is then used to calculate the precision recall. These precision recall values are then plotted and the trapezoidal rule is used to compute the Area Under the Precision Recall Curve. This AUPRC was used as the evaluation metric for the competition and our experiments. All the evaluations for our setup were done using the code provided by the competition organizers and can be found on github [18].

4. Experimental Setup

4.1. Models

The teacher model used in our experiments is the second place model of the DCASE 2019 Task 5 challenge [19]. We received the model weights from the author directly. This model used pretrained weights from the first six layers of the VGGish model [20] as a starting point for transfer learning. The AUPRC of 0.837 for the second place model was good enough for the purposes of our experiments. This model was chosen because the first place model had additional feature augmentation techniques that were not directly relevant to experiments of our novel KD method.

The student model we used is a CNN-LSTM model. The student model has one CNN layer, one LSTM layer, and one fully connected layer. Table 1 shows the model architecture of the student model in more detail. This design choice was motivated by the need to keep the student model as small as possible while still keeping in line with what is conventionally used in Audio Tagging. The first CNN layer was chosen because the top teams in the DCASE 2019 Task 5 challenge use CNN based architectures [21][19][22]. In addition, CNNs have been shown to work well in audio tasks [20] [23]. The LSTM layer was chosen because LSTMs are suited for capturing time series information. More generally, RNNs have had success with audio tasks as well [24] [25]. The hidden dimensions of the LSTM layer in the student model were varied from 128 hidden dimensions to 16 hidden dimensions to test the robustness of the proposed method on different sizes of small models. Table 2 shows the number of parameters for the teacher and student models.

Table 1: *Architecture of CNN-LSTM Model Used*

Layer	Type	Configuration
1	CNN	num_filters: 32 filter_size: 5x5 stride: 2
2	LSTM	hidden_dim: 128, 64, 32, or 16
3	FC	

Table 2: *Number of FLOPS and Parameters in Student and Teacher Models*

		FLOPS (G)	Params (M)
Teacher		14.7	3.96
Student	LSTM 128	0.33	0.742
	LSTM 64	0.17	0.355
	LSTM 32	0.08	0.173
	LSTM 16	0.04	0.086

4.2. Feature Extraction

We followed the same feature extraction process as the second place winner of DCASE 2019 Task 5, matching our teacher model selection. First the audio clips were resampled to $16kHz$. Then log mel-filterbanks were computed with a window of $25ms$ and step size of $10ms$. Each sample has 64 bins so the final feature dimension input to the teacher model is 64×998. Keeping in line with the theme of making the student model as small as possible, the student model features uses 20 bins so the feature dimensions for the student model is 20×998.

4.3. Experiments

There were five setups for our experiments with different $\mathcal{L}_{Total}$ used as the objective function for training: Baseline ‘BCE’, ‘BCE+KD’, ‘BCE+KD+SP’, ‘BCE+KD+IUSP’, and Combination ‘BCE+KD+SP+IUSP’. Where ‘BCE’ is Binary Cross Entropy, ‘KD’ is the original KD paper [1], ‘SP’ is the Similarity Preserving KD [5], and ‘IUSP’ is our proposed method Intra-Utterance Similarity Preserving KD. An example total loss function is shown below:

$$\mathcal{L}_{Total} = \alpha_1 \mathcal{L}_{BCE} + \alpha_2 \mathcal{L}_{KD} + \alpha_3 \mathcal{L}_{SP} + \alpha_4 \mathcal{L}_{IUSP} \quad (7)$$

For all experimental setups, the α hyper parameters were chosen so that all the loss items, $\alpha_1 \mathcal{L}_{BCE}$; $\alpha_2 \mathcal{L}_{KD}$; $\alpha_3 \mathcal{L}_{SP}$; and $\alpha_4 \mathcal{L}_{IUSP}$, were of equal magnitude. The values for $\alpha_{1...4}$ are: 1.0, 10.0, 10.0, and 1.0 respectively. If a particular setup did not include a specific loss item, then the α value for that loss item was 0.0.

The student models were trained using the Adam Optimizer with a learning rate of 0.0001 for 300 epochs. There was also a early stopping criteria where training stops if the AUPRC of the validation set did not improve for more than 20 epochs. In practice, most of the training plateaued around 50 epochs.

4.4. Tuning Intermediate Hint Layers

Given that the teacher and student models have significantly different architectures, there are many different choices when choosing potential intermediate outputs to compare. Referring to Figure 2, for the teacher model, we used the outputs after the Max Pooling layers as potential hint layers to guide the student model. For the student model, there were only three layers in total so there are only two possible intermediate hint layers to use. Four potential hint layers from the teacher model and two potential hint layers from the student model means that we tried eight different combinations of $(A^{(l)}_{Teacher}, A^{(l)}_{Student})$ pairs that can be used to calculate the SP and IUSP losses.

Four trials each were performed on the validation dataset with all possible combinations and for all LSTM sizes. The configuration with the best average micro AUPRC across all LSTM sizes was chosen for final results and analysis. For ‘BCE+KD+SP’, we use the intermediate output from the second Max Pooling layer of the teacher and the intermediate output from the CNN layer of the student. For ‘BCE+KD+IUSP’, we use the intermediate output from the first Max Pooling layer of the teacher and the intermediate output from the CNN layer of the student. For ‘BCE+KD+SP+IUSP’, both ‘SP’ and ‘IUSP’ used the intermediate output from the second Max Pooling layer of the teacher and the intermediate output of the CNN layer from the student.

5. Results and Analysis

5.1. Top Level Results

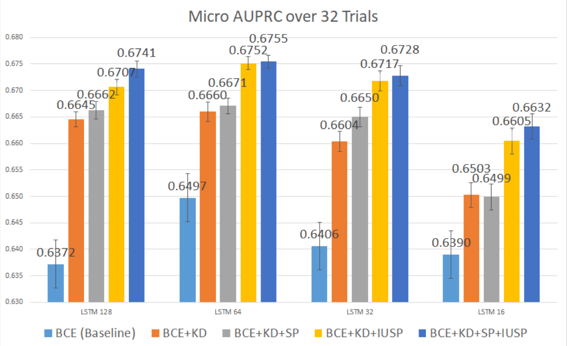

Figure 3: *Overall Micro AUPRC for each Student Model with standard error bars. The higher the AUPRC the better.*

A full graph of the micro AUPRC results is shown in Figure 3 with standard error of the mean computed from 32 trials of each experimental setup. As mentioned before, the teacher model has a micro AUPRC of 0.837. In our experiments, the dark blue bar, the setup ‘BCE+KD+SP+IUSP’, has the highest micro AUPRC over 32 trials across all different model sizes. The relative improvements over the Baseline and

'BCE+KD+SP' is pretty consistent throughout all the varying LSTM dimensions. It seems that there is an overall trend that adding 'SP' to the experimental setups improves results. However, any improvement is small and the standard error for setups with and without 'SP' overlaps.

Table 3 shows the relative and absolute improvements. The addition of our KD method, 'BCE+KD+SP+IUSP', provides a 27.1% to 122.4% increase in performance compared to Similarity Preserving KD, 'BCE+KD+SP'.

Table 3: *Relative and Absolute Improvements of Various KD Methods for different LSTM hidden dimensions.*

	128	**64**	**32**	**16**
Micro AUPRC				
BCE	0.637	0.650	0.641	0.639
BCE+KD+SP	0.666	0.667	0.665	0.650
BCE+KD+SP+IUSP	**0.674**	**0.675**	**0.673**	**0.663**
Improvement over Baseline BCE				
BCE+KD+SP	0.029	0.017	0.024	0.011
BCE+KD+SP+IUSP	0.037	0.026	0.032	0.024
Percent Increase	27.1%	48.2%	31.9%	122.4%

5.2. Class Level AUPRC Results

Table 4 is of the class-wise results for the models with LSTM 128. The full results for all five setups are not shown because of space constraints.

Table 4: *Class-wise AUPRC. Setup 1 is 'BCE+KD', Setup 2 is 'BCE+KD+SP', Setup 3 is 'BCE+KD+IUSP', and Setup 4 is 'BCE+KD+SP+IUSP'*

	Setup 1	Setup 2	Setup 3	Setup 4
LSTM 128				
engine	0.7938	0.8038	0.8009	**0.8147**
machinery -impact	0.5081	0.5021	0.5606	**0.5659**
non-machinery -impact	0.0941	0.0889	0.0955	**0.1071**
powered-saw	0.4394	0.4477	**0.4522**	0.4429
alert-signal	0.6033	0.6164	0.6112	**0.6367**
music	**0.1564**	0.1343	0.1532	0.1074
human-voice	**0.7357**	0.7255	0.7280	0.7119
dog	0.6422	**0.6438**	0.6360	0.6427

The best method for each category is mostly consistent throughout all the different LSTM hidden dimensions. Figure 4 shows the difference in class-wise AUPRC of the combination 'BCE+KD+SP+IUSP' system over the Baseline 'BCE+KD' system for different LSTM hidden dimensions. There is improvement in some categories and degradation in others. The overall improvement in AUPRC is likely dataset dependent.

The improvements seen are in line with what is expected given the preservation of Intra-Utterance Similarity. Events like 'machinery-impact' and 'non-machinery-impact' are likely loud, singular sounds. Events like 'alert-signal' are likely loud, repetitive sounds. Both of these types of events would have obvious indications in the spectrogram. So 'BCE+KD+SP+IUSP' performs the best in all the aforementioned categories.

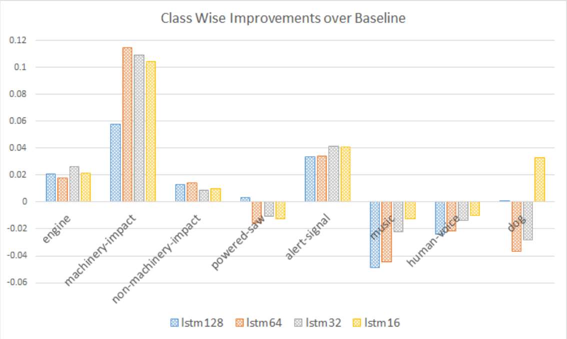

Figure 4: *The class-wise improvements of the combination 'BCE+KD+SP+IUSP' system over the baseline 'BCE+KD' system. This is the difference in class-wise AUPRC.*

Going back to the spectrogram and Intra-Utterance Similarity matrix of Figure 1, the pattern that is enforced is very clear. Things like 'music' and 'human-voice' are a lot more varied in terms of frequency in the spectrogram. So in general, there is no meaningful 'IUSP' to be learned. An example spectrogram of the 'human-voice' category is shown in Figure 5.

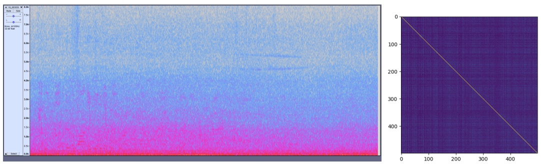

Figure 5: *Spectrogram and 'IUSP' Matrix for 'human-voice' category.*

This hypothesis that our proposed method only significantly improves results for strong stationary signals or singular high energy events is bolstered by the fact that we did not see any improvements on Audio Scene Classification. We tried using our proposed method on the DCASE 2019 Task 1 [26] with a top team [27] as the teacher model. The different scene classes were of public areas such as 'metro', 'bus-station', and 'shopping-mall'. The experimental setup and procedure used for Task 1 was the same as the one used for DCASE 2019 Task 5. However, when comparing the results there was no clear difference between all 5 setups. In terms of class-wise performance, there was also no clear pattern to the class-wise improvements or degradations.

6. Conclusion

In conclusion, our proposed KD method, Intra-Utterance Similarity Preserving KD, shows improvement over Similarity Preserving KD. This is true for both setups 'BCE+KD+IUSP' and 'BCE+KD+SP+IUSP', though the combination of all the loss items performs the best. There is a 27.1% to 122.4% percent increase in performance. Experimental results indicate that this method performs best when the dataset contains many audio clips of sounds with strong repetitions or loud singular events. Further experiments on a wider range of classes and datasets may yield a better heuristic in determining the usefulness of 'IUSP'.

7. References

[1] G. E. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *ArXiv*, vol. abs/1503.02531, 2015.

[2] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, "Fitnets: Hints for thin deep nets," *CoRR*, vol. abs/1412.6550, 2015.

[3] Y. Tian, D. Krishnan, and P. Isola, "Contrastive representation distillation," *ArXiv*, vol. abs/1910.10699, 2020.

[4] J. Yim, D. Joo, J. Bae, and J. Kim, "A gift from knowledge distillation: Fast optimization, network minimization and transfer learning," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[5] F. Tung and G. Mori, "Similarity-preserving knowledge distillation," *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1365–1374, 2019.

[6] S. Zagoruyko and N. Komodakis, "Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer," *ArXiv*, vol. abs/1612.03928, 2017.

[7] W. Park, D. Kim, Y. Lu, and M. Cho, "Relational knowledge distillation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[8] A. Krizhevsky, "Learning multiple layers of features from tiny images," Tech. Rep., 2009.

[9] Y. Chebotar and A. Waters, "Distilling knowledge from ensembles of neural networks for speech recognition," in *Interspeech 2016*, 2016, pp. 3439–3443. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2016-1190

[10] J. Kim, M. El-Khamy, and J. Lee, "Bridgenets: Student-teacher transfer learning based on recursive neural networks and its application to distant speech recognition," *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5719–5723, 2018.

[11] L. Lu, M. Guo, and S. Renals, "Knowledge distillation for small-footprint highway networks," *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4820–4824, 2017.

[12] J. Li, R. Zhao, J.-T. Huang, and Y. Gong, "Learning small-size dnn with output-distribution-based criteria," in *Interspeech*, September 2014. [Online]. Available: https://www.microsoft.com/en-us/research/publication/learning-small-size-dnn-with-output-distribution-based-criteria/

[13] B. Shi, M. Sun, C.-C. Kao, V. Rozgic, S. Matsoukas, and C. Wang, "Compression of acoustic event detection models with quantized distillation," in *INTERSPEECH*, 2019.

[14] A. Kumar and V. K. Ithapu, "Secost: Sequential co-supervision for weakly labeled audio event detection," *ArXiv*, vol. abs/1910.11789, 2019.

[15] B. Shi, M. Sun, C. Kao, V. Rozgic, S. Matsoukas, and C. Wang, "Semi-supervised acoustic event detection based on tri-training," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 750–754.

[16] J.-w. Jung, H.-S. Heo, H.-j. Shim, and H.-J. Yu, "Knowledge distillation with specialist models in acoustic scene classification," DCASE2019 Challenge, Tech. Rep., June 2019.

[17] H.-S. Heo, J. weon Jung, H. jin Shim, and H.-J. Yu, "Acoustic scene classification using teacher-student learning with soft-labels," in *INTERSPEECH*, 2019.

[18] 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.

[19] B. Kim, "Convolutional neural networks with transfer learning for urban sound tagging," DCASE2019 Challenge, Tech. Rep., September 2019.

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

[21] S. Adapa, "Urban sound tagging using convolutional neural networks," DCASE2019 Challenge, Tech. Rep., September 2019.

[22] L. Cui, S. Ji, X. Han, and J. Wang, "Time-frequency segmenation attention neural network for urban sound tagging," DCASE2019 Challenge, Tech. Rep., September 2019.

[23] N. Takahashi, M. Gygli, B. Pfister, and L. V. Gool, "Deep convolutional neural networks and data augmentation for acoustic event recognition," in *INTERSPEECH*, 2016.

[24] Y. Xu, Q. Kong, W. Wang, and M. D. Plumbley, "Large-scale weakly supervised audio classification using gated convolutional neural network," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 121–125.

[25] G. Parascandolo, H. Huttunen, and T. Virtanen, "Recurrent neural networks for polyphonic sound event detection in real life recordings," *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6440–6444, 2016.

[26] 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

[27] K. Koutini, H. Eghbal-zadeh, and G. Widmer, "Acoustic scene classification and audio tagging with receptive-field-regularized CNNs," DCASE2019 Challenge, Tech. Rep., June 2019.