SEMI-SUPERVISED ACOUSTIC EVENT DETECTION BASED ON TRI-TRAINING

Bowen Shi[1], Ming Sun[2], Chieh-Chi Kao[2], Viktor Rozgic[2], Spyros Matsoukas[2], Chao Wang[2]

[1]Toyota Technological Institute at Chicago
[2]Amazon
bshi@ttic.edu, {mingsun,chiehchi,rozgicv,matsouka,wngcha}@amazon.com

ABSTRACT

This paper presents our work of training acoustic event detection (AED) models using unlabeled dataset. Recent acoustic event detectors are based on large-scale neural networks, which are typically trained with huge amounts of labeled data. Labels for acoustic events are expensive to obtain, and relevant acoustic event audios can be limited, especially for rare events. In this paper we leverage an Internet-scale unlabeled dataset with potential domain shift to improve the detection of acoustic events. Based on the classic tri-training approach, our proposed method shows accuracy improvement over both the supervised training baseline, and semi-supervised self-training set-up, in all pre-defined acoustic event detection tasks. As our approach relies on ensemble models, we further show the improvements can be distilled to a single model via knowledge distillation, with the resulting single student model maintaining high accuracy of teacher ensemble models.

Index Terms— acoustic event detection, tri-training, semi-supervised learning

1. INTRODUCTION

Acoustic event detection (AED) is the task of detecting whether certain events occur in an audio clip. It can be applied in many areas such as surveillance [1, 2], and recommendation systems [3]. Conventionally AED has been addressed with automatic speech recognition techniques, e.g. with features such as mel-frequency cepstrum coefficients (MFCC) and classifiers based on hidden markov model (HMM). In recent years, with the advances in speech recognition [4] and image recognition [5] as well as size increasing of datasets [6][7], there are more deep learning based approaches applied to tackle AED tasks. For instance, the recently proposed Audioset [6] comprises 1,789,621 10-second audio segments from a wide domain of 632 categories. Convolutional neural network (CNN) [8, 9] or CNN-based approaches (e.g convolutional recurrent neural network [10, 11]) are used and have shown improvements over traditional approaches. Though accuracy has been much improved in many AED tasks, state-of-the-art models often requires large number of labeled training data. Labeled data can be be quite limited under certain scenarios (e.g., for rare events [12]). The focus of this paper is on leveraging unlabeled audios to improve accuracy for AED.

Our main contributions include the following: (1). We propose an ensemble method based on the classic tri-training that shows improvements in all acoustic events we investigate in a realistic semi-supervised setting (Internet-scale unlabeled dataset with domain shift) (2). We show the improvements of the ensembled models can be distilled into a single model via knowledge distillation. As a result, there is no increase of computational costs during inference.

2. RELATED WORK

There has been a great volume of work on semi-supervised learning. A broad class of approaches contains feature learning with unlabeled data, based on generative models including variational autoencoders [13], or generative adversarial networks [14, 15]. Another category of semi-supervised learning approaches is based on achieving certain smoothness effects with unlabeled data. For example, virtual adversarial training [16] relies on smoothing model training with an regularization term based on adversarial direction. Those semi-supervised models are often evaluated in a 'simulated' setting by discarding many labels from an existing large labeled dataset, and they are sensitive to class distribution mismatch [17]. Instead, our approach belongs to the family of bootstrapping methods, where models are often treated as a black box to assign pseudo labels on unlabeled data. Self-training is the simplest one of such category, which refers to retraining a model based on its own predictions on unlabeled data. Despite its simplicity, self-training has been widely applied in practice. [18] proposed an approach based on self-training to visual structure prediction problems. For AED, self-training is employed to perform semi-supervised learning from Youtube audios [19]. Compared to previous efforts, our method is simpler but effective. The whole method can be directly built on audio features, e.g. log mel-filter bank energies (LFBEs), without involving complex data augmentation steps as in [18, 20]. Our experiments are placed in a realistic setting where unlabeled data come from Internet, and form a dissimilar distribution

from the labeled dataset.

3. METHODS

In this section we describe the methods we use for semi-supervised learning. We focus on a multi-event classification setting. Given an audio signal I (e.g. LFBE), the task is to train a model $\mathbf{f}$ to predict a multi-hot vector $\mathbf{y} \in \{0,1\}^C$ with C being the size of event set $\mathcal{E}$ and y_c being a binary indicator whether event c is present in I. Note the prediction $\mathbf{f}(I)$ is not a distribution over event set $\mathcal{E}$ since multiple events can occur in I. We denote $\mathcal{D}_L = \{(I, \mathbf{y})\}$ as the labeled datset and $\mathcal{D}_{UL} = \{I\}$ as the unlabeled dataset. In supervised setting, we train model $\mathbf{f}$ using cross-entropy loss (see equation 1), where w_c is the penalty of positive mis-classification of class c. w_c is tuned to balance losses computed from positive and negative instances.

$$L = -\sum_{(I,\mathbf{y})\in\mathcal{D}_L}\sum_{c=1}^{C}\{w_c y_c \log f_c(I) + (1-y_c)\log(1-f_c(I))\} \tag{1}$$

Self-training is a natural heuristic, which leverages a trained model to make predictions on unlabeled data and uses resulting pseudo labels to update the model . More formally, self-training consists of the following iterative process. A model $\mathbf{f}$ is initially trained with minimizing loss defined in equation 1 with labeled data $\mathcal{D}_L$. At each iteration, we assign probability $\mathbf{p}(x) \in \mathbb{R}^C$ to every unlabeled example $x \in \mathcal{D}_{UL}$ by applying model $\mathbf{f}$. Top k unlabeled data are selected for each class and added to the labeled dataset L based on class score $m_c(\cdot)$, $\forall c \in \mathcal{C}$. Model $\mathbf{f}$ is re-trained with labeled dataset L' augmented with kC examples from $\mathcal{D}_{UL}$. Instead of directly setting a threshold for selecting data, we sort and select examples from $\mathcal{D}_{UL}$. As the model is applied on a different dataset $\mathcal{D}_{UL}$ with inevitable domain shifts at test time, relative order of confidence is more reliable than the absolute value of probabilty $\mathbf{p}$.

Tri-training One flaw of self-training is that the mistakes made by the model can be amplified by adding errorneous data. To avoid this, we can train multiple models and add data according to the agreement of those models. In tri-training [21], we first train three models $\mathbf{f}_1$, $\mathbf{f}_2$, $\mathbf{f}_3$ independently. New data is added to train a particular model if other two models agrees on its label. In our multi-binary classification setting specifically, we train three different models by bootstrapping the training set. Data x is considered as a pseudo-label candidate for class c of one model if its probability output by other two models are sufficiently high. We select top k pseudo-labeled candidate data according to its score for each class (e.g. average probability of other two models) (see Algorithm 1). The data augmentation process is repeated for certain number of iterations. In classic tri-training, the final three models are ensembled during test time. However, in many real-world applications, the data distribution of unlabeled dataset can vary from labeled dataset, where the test set is from. Thus we ensemble models augmented through tri-training scheme, as well as the initial models trained with bootstrapped labeled data. As a results, there are in total 6 models ensembed, as shown at the end of Algorithm 1.

Algorithm 1: Ensemble-based tri-training

```
Initialize;
for i ∈ {1,2,3} do
    D_i^0 = bootstrap(D_L);
    Train f_i^0 using eq.1 with D_i^0;
end
for t ∈ {1,...,T} do
    for i ∈ {1,2,3} do
        D_i^t ← ∅;
        for x ∈ D_UL do
            P_c ← ∅;
            for c ∈ {1,...,C} do
                if f_jc^{t-1}(x) > θ_c ∧ f_hc^{t-1}(x) > θ_c(j ≠
                 h ≠ i) then
                    P_c ← P_c ∪ {(x, (f_jc^{t-1}(x)+f_hc^{t-1}(x))/2)};
                end
                D_i^t ← D_i^t ∪ top-k(P_c);
            end
        end
        Train f_i^t using eq.1 with ⋃_{t'=0}^{t} D_i^{t'};
    end
end
Ensemble f_i^T and f_i^0(i ∈ {1,2,3});
```

In our tri-training scheme, we need to rely on ensemble of models for test. In reality this can take a lot of time and induce heavy memory and computation burdens, particularly for resource-constraint applications. Thus we propose to "trasnfer" the ensembled models into a single model with knowledge distillation. Specifically, we train a single model $\mathbf{f}^s$ (student) to mimic its output distribution to the ensembled model $\mathbf{f}^e$ by minimizing loss 2 adapted from the commonly used single-class classification setting [22].

$$\begin{aligned}
L_{kd} &= \sum_{(I,\mathbf{y})\in\mathcal{D}_L}\{\alpha T^2 l(I, \mathbf{f}^e(T)) + (1-\alpha) l(I, \mathbf{y})\} \\
l(I, \mathbf{y}') &= \sum_{c=1}^{C}\{w_c y'_c \log f_c^s(I) + (1-y'_c)\log(1-f_c^s(I))\} \\
\mathbf{f}^e(T) &= \frac{1}{1+\exp(-\frac{\mathbf{z}}{T})}
\end{aligned} \tag{2}$$

, where $\mathbf{z}$ is the logits of ensembled model. T and α are hyperparameters controlling the softness of teacher logits $\mathbf{z}$ and

relative weight of distillation loss, respectively. For ensembling, we average the probabilities output by individual models and convert it back to logits. Note that only labeled dataset $\mathcal{D}_L$ is used for training single student model.

4. EXPERIMENTS

4.1. Experimental Setting

Data The labeled dataset we use is a subset from Audioset [6]. In particular, we select dog sound, baby crying and gunshots as the target events, which include both human and non-human vocals, as well as different durations of sound events. The three events included in Audioset amount to 13,460, 2,313 and 4,083 respectively, and we use all of them. All the three events are often considered rare events where number of labeled examples are quite limited in many real-world application scenarios. Note the class of dog contains any sounds produced by dog (e.g., barking, yipping), which makes the intra-class variation much bigger compared to other two events.

In addition to the three events, we randomly selected 36,036 examples from all other audio clips in Audioset as negative samples. The negative vs. positive ratio is high especially for baby crying and gunshots (> 10), which also aims to simulate the scarcity property of those rare events. We randomly split the whole subset for training (70%), validation (10%) and test (20%). Additional efforts has been made to ensure the distribution of events roughly same across different sets.

We use Amazon Instant Video (AIV) as our unlabeled dataset. The AIV data is a collection of audio parts of Amazon instant videos. To be consistent with Audioset, we split AIV audios into 10-second segments and the amount is 5,404,106 in total. Note the domain difference between AIV and Audioset can be large because AIV set is mainly media sounds (e.g. from films and TV shows) while the latter one contains many audio clips taken in real life.

Implementation details We first compute LFBE features for each audio clip. It is calculated with window size of 25 ms and hop size of 10 ms. The number of mel coefficients is 64, which gives us log-mel spectrogram feature of size 998×64 for each audio clip. Features are further normalized by global cepstral mean and variance normalization (CMVN).

We use DenseNet [23] with 63 layers as our backbone model. The DenseNet we use contains 4 dense blocks with respectively 3, 6, 12 and 8 dense layers, where each layer is composed of batch normalization, ReLU, 1×1 convolution, batch normalization, ReLU and 3×3 convolution. The choice of model is based on dev performance under fully-supervised setting. As we have to run inference on large amount of unlabeled data, inference speed is also an important factor along with the accuracy. Our experimented models include ResNet [24], DenseNet and Conv-RNN [25, 26] with different layers, which are among the state-of-the-art models for acoustic event detection. According to our experiments, DenseNet-63 achieves highest accuracy and also has relative small inference latency.

For ensemble-based tri-training, we pick the top 5,000 data for each class following algorithm 1. The number is tuned with dev set and will be analyzed in following sections. Model is re-trained from scratch when pseudo-labeled data are added. The tri-training process is repeated for one iteration. Here we did not observe further improvement by taking more iterations. For all experiments we use Adam optimizer with learning rate of 0.001 and batch size of 64. We tuned penalty on positive loss (w_c) on dev set and found setting it to be the ratio between positive and negative examples of each class gives overall best results. This also prevents from tuning w_c for every class under different settings.

Evaluation Metric We evaluate the performance of models based on area under curve (AUC) and equal error rate (EER) on detection error tradeoff (DET) curve (vertical: false negative rate (FNR), horizontal: false positive rate (FPR)). Performance is measured for individual events.

Baselines We compare ensemble based tri-training with the following two baselines. (1) Fully-supervised DenseNet-63, and (2) Self-training. Both baseline model training follows the same experimental setting as tri-training, but with a single DenseNet-63.

4.2. Results

The results of different models are shown in table 1. The proposed ensemble-based tri-training outperforms other semi-supervised learning approaches in all three tasks. Detailed analysis on the improvements will be presented in the following analysis section. In principle, semi-supervised approaches should be lower-bounded by the fully-supervised baseline. But as we evaluate classes individually, it is possible to have degradation for some classes. For gunshots, all semi-supervised approaches improves over the supervised baseline. This may be related to the small domain discrepancy between unlabeled and labeled datasets for this particular event. We find many gunshot audioclips in Audioset are from multi-media source (e.g. video games), which is similar to the unlabeled AIV data. This shows semi-supervised learning helps especially when labeled and unlabeled data are from same domain.

The results of knowledge distillation trained DenseNet-63 (Tri-KD) with ensemble of tri-training models as teacher are also listed in table 1. Though there is small degradation compared to the tri-training, it outperforms the other single models. The improvement shows that it is possible to distill the gain brought by using large amount of unlabeled data into a single supervised-trained model, so that there are no additional computational costs during inference.

Ablation Study There are three factors contributing to the

Event	AUC (%)				EER (%)			
	Sup_s	Self_s	Tri_e	Tri-KD_s	Sup_s	Self_s	Tri_e	Tri-KD_s
Dog	4.32	4.42	**3.26**	3.49	11.11	11.07	**9.29**	9.80
Baby-cry	2.20	2.89	**1.42**	1.69	6.56	7.34	**5.41**	6.01
Gunshots	2.07	1.77	**1.31**	1.51	6.41	5.78	**4.70**	5.41

Table 1. Performance of models (on test set). Lower is better. Sup: fully-supervised baseline, Self: self-training, Tri: ensemble-based tri-training, Tri-KD: distilled model from ensembled tri-training models, e: Ensembled model, s: Single model

performance improvements for semi-supervised training: the scale of model, unlabeled data, ensemble of models. We further study the effects of these factors on the models and results are shown in table 2. To avoid tuning on test set, all analysis are done on dev set. Compared to other approaches, tri-training has more diversity as three models are trained by bootstrapping the original training set. This increased scale brings improvements over the baseline even without any unlabeled data. Adding unlabeled data improves the performance in general despite small degradation of EER on baby crying. In binary classification on imbalanced data, adding pseudo-labeled data balances the training set and we observed model converged much faster compared to supervised baseline. The unapparent improvement on baby crying is related to the domain difference between the unlabeled and labeled datasets. We also observe that simple ensemble of models trained with and without unlabeled data can mitigate the side-effects of using unlabeled data brought by domain discrepancy.

Event	AUC (%)				EER (%)			
	Sup	+Ens	+Ens +Data	+2xEns +Data	Sup	+Ens	+Ens +Data	+2xEns +Data
Dog	4.48	3.96	3.29	**3.28**	11.06	9.81	9.09	**8.95**
Baby-cry	2.89	2.86	2.75	**2.57**	8.30	**7.95**	8.71	8.21
Gunshots	2.46	1.53	1.39	**1.28**	7.68	6.11	5.41	**5.22**

Table 2. How different factors contributes to the performance of tri-training (on dev set). Lower is better. Sup: supervised baseline, +Ens: ensembled 3 models trained with only labeled data, +Ens+Data: ensembled tri-training models with unlabeled data, 2xEns+Data: ensembled tri-training models with and without unlabeled data. Note that Sup in table 2 and Sup_s in table 1, 2xEns+Data in table 2 and Tri in table 1 refer to same approach.

Varying amount of pseudo-labeled data Number of pseudo-labeled data to add is an important hyper-parameter to tune. Table 3 summarizes our analysis on this front. Adding more pseudo-labeled data raises the percentage of data with wrong labels in training set. We find that within certain range, the side-effects brought by the noisy data can be compensated by the data amount. Adding few data with high confidence is not as effective, because those are mainly "easy" data with which models are not guaranteed to be strengthened. Varying the data amount does not have as much impact on baby crying as other two events, which may be related to the domain shift of this particular event in unlabeled dataset.

Event	AUC (%)			EER (%)		
	1k	5k	10k	1k	5k	10k
Dog	3.82	**3.28**	4.02	10.02	**8.95**	10.46
Baby-cry	2.69	**2.57**	2.82	**8.20**	8.21	8.73
Gunshots	2.09	**1.28**	1.75	6.46	**5.22**	6.16

Table 3. How number of pseudo-labeled data impact performance (of Tri in table 1, on dev set). Lower is better. Our experimental results described earlier are based on 5k.

Varying size of labeled training set In our default experimental setting, we have a relatively larger training set (ratio of train set to test set = 3.5). To see how the model performs with different size of training set, we reduced number of training data. Specifically we keep the same test and validation set and change the ratio between training and test set to $\{0.5, 1.0, 2.0, 3.5\}$ (3.5 is the whole original training set). According to figure 1, our semi-supervised learning approach shows consistent gains with different size of training set, and using unlabeled data brings more gain when training set is relatively smaller.

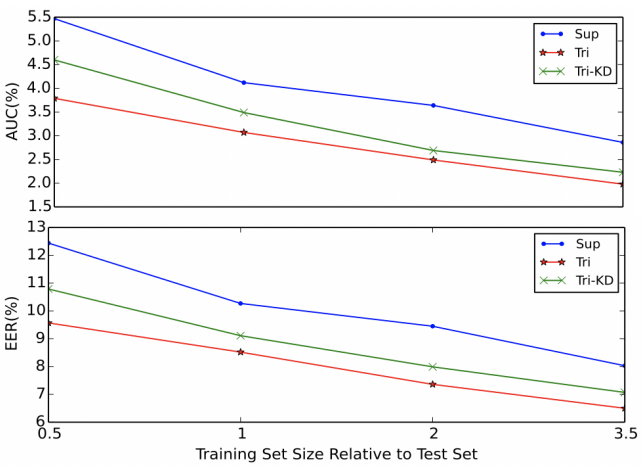

Fig. 1. Average AUC (%) and EER (%) (on test) of three events with different training set to test set ratio. Training dataset is sampled according to each train vs. test ratio

5. CONCLUSIONS

We investigate using large number of unlabeled data to improve acoustic event detection. Our proposed approach which is based on classic tri-training with ensembling shows consistent improvements over models trained with labeled data, as well as with self-training. In addition, we show that such improvements brought by the ensembled tri-training models can be distilled into a single model, which shows improved accuracy with same computational cost during inference.

6. REFERENCES

[1] M. Cristani, M.Mecego, and V.Murino, "Audio-visual event recognition in surveillance video sequences," *IEEE Transactions on Multimedia*, vol. 9, pp. 257–267, 2 2007.

[2] G. Valenzise, L.Gerosa, M. Tagliasacchi, F. Antonacci, and A. Sarti, "Scream and gunshot detection and localization for audio-surveillance systems," in *IEEE Conference on Advanced Video and Signal Based Surveillance, AVSS*, 2007, pp. 21–26.

[3] P. Cano, M.Koppenberger, and N. Wack, "Content-based music audio recommendation," in *ACM International Conference on Multimedia*, 2005, pp. 211–213.

[4] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 11 2012.

[5] A. Krizhevsky, I. Sutskever, and G.E.Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012.

[6] J. F. Gemmeke, D.P.W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *ICASSP*, 2017.

[7] H. Fan, J. Zhou, and C. Fuegen, "Facebook acoustic events dataset," in *ICASSP*, 2018.

[8] Naoya Takahashi, Michael Gygli, Beat Pfister, and Luc Van Gool, "Deep convolutional neural networks and data augmentation for acoustic event detection," *CoRR*, 2016.

[9] 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," in *ICASSP*, 2017.

[10] Y. Xu, Q. Kong, Q. Huang, W. Wang, and M. D. Plumbley, "Convolutional gated recurrent neural network incorporating spatial features for audio tagging," in *IJCNN*, 2017.

[11] C-C. Kao, W. Wang, M. Sun, and C. Wang, "R-crnn: Region-based convolutional recurrent neural network for audio event detection," in *Interspeech*, 2018.

[12] V. Arora, M. Sun, and C. Wang, "Deep embeddings for rare audio event detection with imbalanced data," in *ICASSP*, 2019.

[13] D. Kingma, S. Mohamed, D.J. Rezende, and M. Welling, "Semi-supervised learning with deep generative models," in *NIPS*, 2014.

[14] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," in *NIPS*, 2016.

[15] J.T. Springenberg, "Unsupervised and semi-supervised learning with categorical generative adversarial networks," in *ICLR*, 2015.

[16] M. Takeru, M. Shinichi, K. Masanori, and I. Shin, "Virtual adversarial training: a regulariza- tion method for supervised and semi-supervised learning," arXiv:1704.03976, 2017.

[17] A. Oliver, A. Odena, C. Raffel, E.D. Cubuk, and I. Googfellow, "Realistic evaluation of semi-supervised learning algorithms," arxiv:1804.09170, 2018.

[18] I. Radosavovic, P. Dollar, R. Girshick, G. Gkioxari, and K. He, "Data distillation: Towards omni-supervised learning," in *CVPR*, 2018.

[19] B. Elizalde, A. Shah, S. Dalmia, M.H. Lee, R. Badlani, A. Kumar, B. Raj, and I. Lane, "An approach for self-training audio event detectors using web data," in *EUSIPCO*, 2017.

[20] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *NIPS*, 2017.

[21] Z-H. Zhou and M. Li, "Tri-training: Exploiting unlabled data using three classifiers," *IEEE Trans.Data Eng*, vol. 17, no. 11, pp. 15291541.

[22] G. E. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *CoRR*, 2015.

[23] G. Huang, Z. Liu, L.V.D Maaten, and K. Weinberger, "Densely connected convolutional networks," in *CVPR*, 2017.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[25] Y. Xu, Q. Kong, Q. Huang, W. Wang, and M. D. Plumbley, "Convolutional gated recurrent neural network incorporating spatial features for audio tagging," in *IJCNN*, 2017.

[26] E. Cakir, G. Parascandolo, T. Heittola, H. Huttunen, and T. Virtanen, "Convolutional recurrent neural networks for polyphonic sound event detection," arXiv preprint arXiv:1702.06286, 2017.