Deep CNN Framework for Audio Event Recognition using Weakly Labeled Web Data

Anurag Kumar, Bhiksha Raj
Language Technologies Institute, School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
alnu@andrew.cmu.edu, bhiksha@cs.cmu.edu

Abstract

The development of audio event recognition models requires labeled training data, which are generally hard to obtain. One promising source of recordings of audio events is the large amount of multimedia data on the web. In particular, if the audio content analysis must itself be performed on web audio, it is important to train the recognizers themselves from such data. Training from these web data, however, poses several challenges, the most important being the availability of labels : labels, if any, that may be obtained for the data are generally *weak*, and not of the kind conventionally required for training detectors or classifiers. We propose a robust and efficient deep convolutional neural network (CNN) based framework to learn audio event recognizers from weakly labeled data. The proposed method can train from and analyze recordings of variable length in an efficient manner and outperforms a network trained with *strongly labeled* web data by a considerable margin.

1 Introduction

In last few years audio event classification and detection (AEC) has become an important research problem in the broad area of Machine Learning and Signal Processing. It has applications in surveillance [2], content-based indexing and retrieval of multimedia data [16, 17] and human computer interaction [9] to name a few. Multimedia content analysis for indexing and retrieval is perhaps the most important one, given that the amount of multimedia data on the web is growing at an exponential rate. However, unlike its visual counterpart, AEC has not yet moved up to very large scale modeling in terms of both training data size and number of sound events. For vision, most large-scale image datasets have been collected from the web [3, 15]. Multimedia data on the web are an important source of audio events as well. However, the recordings are usually unlabeled; to use them to train models for audio events one must label them with the time stamps that mark the temporal boundaries of the occurrences of different audio events.

This form of labels, often referred to as "strong" labels, is a major bottleneck for large scale AEC. Creating such labeled *audio* data is very difficult. Moreover, marking beginning and ends of sound events can be inherently ambiguous, which makes the task even harder [8].

To address the problems of working with strongly labeled data, [7] proposed that audio-event detects may be trained using *weakly* labeled data. In weakly labeled data only the presence or absence of the events is indicated; their actual location within the recordings are not marked. Since time stamps need not be marked it is much easier to label a recording for audio events. Moreover, the event when present in the recording is available to the learner in its entire natural form, without any assumption or bias about the beginnings and ends which might get introduced by the annotator. Hence, the learning algorithm interprets the data in a more natural way.

In this work we argue that weak-label based learning approaches for AEC gives us ways to exploit the vast and growing amount of multimedia data on the web; which as we stated before are a rich source of audio events. One can use the metadata associated with the multimedia recordings to automatically generate weak labels for them. Video-sharing websites such as *Youtube* allows one to easily collect large amounts of weakly labeled data for any given audio event. Hence, using web data an AEC framework can be developed without requiring explicit labeling effort. This forms the

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

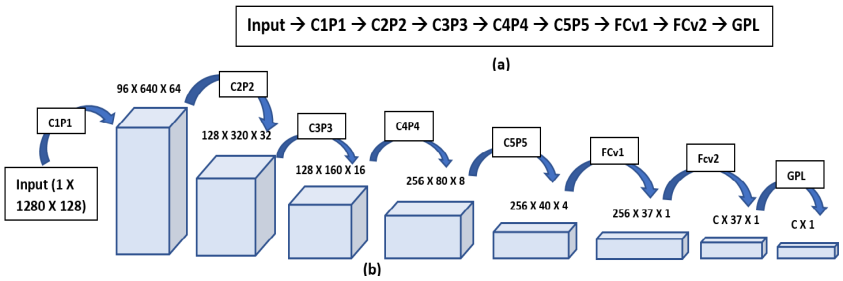

Figure 1: (a) Weakly Labeled CNN, $\mathcal{N}_W$. GPL is global pooling (max here) layer. (b) Shows network output shapes for an input with 1280 frames in mel spectra ($\sim$ 15 second long recording)

primary motivation of this work where we aim to use web data for training audio event recognition models. A second motivation is that in order to be able to successfully deploy AEC models for content-based indexing and retrieval of multimedia, and other real world applications, it is perhaps *essential* to train models on these data.

Learning from web data poses several challenges. Web multimedia data are primarily consumer generated and are generally noisy. The weak labels may encompass long audio segments, whereas the audio event of interest may itself be very short. Often, several sounds events overlap, unlike the exemplars found in several of the audio events datasets such as [12].

Hershey *et al.* [6] used audio from Youtube videos to train and compare different well known Convolutional Neural Network (CNN) architectures for AEC. Although they acknowledge the web data as being weakly labeled, the CNN models are trained in a fully supervised manner under the assumption that the labels are, in fact, strong. We call this method *Strong Label Assumption Training* (SLAT). More specifically, audio recordings are chunked into small fixed-length segments to train the CNN model. The labels of these segments are taken to be the same as the recording-level labels – in essence, the event is assumed to be present through out the recording. This, however, is not an efficient approach as it can result in significant amount of label noise. An audio event, say *door bell ringing*, may be present for only a few seconds in a recording which may be several minutes long, a fact that is ignored in assuming that the label is strong. Moreover, segment wise training of CNN is cumbersome and computationally inefficient. Segmenting recordings is a preprocessing step in itself, and often one must experiment with different segment sizes, which would require repeated preprocessing of the data. Also, a significant portion of computational operations done by the network are common over different segments and in segment-wise training these operations are repeated.

In this paper we propose a CNN-based framework which treats weak labels as *weak* and factors in this information during training. Our experimental results show that this is much superior to training done under the strong-label assumptions. Moreover, our proposed framework can process input recordings of variable length, which makes the training and test process more efficient and convinient. The network design does not require a fixed size input and hence segmentation as a preprocessing step is not needed. The segmentation of audio recordings is automatically done by the network and the network design controls the segment and hop size. Moreover, it is also computationally efficient as common operations are not repeated.

2 CNN For Weakly Labeled Audio

Before going into the details of CNN models, we describe the audio features used to train them. Log mel spectra are used as acoustic features. We employ 4 different FFT sizes (23ms (1024), 11.5ms (512), 46ms (2048), and 92ms (4096)) to extract multi-scale mel-spectral features. It can also be thought of as a data augmentation method. The sampling rate for all recordings is 44100 Hz. The hop size is fixed to 512 for all four window sizes and 128 mel-bands are used to extract mel spectra.

Our proposed CNN network, $\mathcal{N}_W$, for AEC using weakly labeled audio is shown in Figure 1 along with an example of output shape after each layer for a recording with 1280 frames in mel spectra. C1P1 to C5P5 are covolutional layers followed by max-pooling layers. The filter size in all convolutional layers are 3×3, and stride and padding values are 1 in all cases. ReLU activation is used for all convolutional layers. Rest of the characteristics of the network are explained under following subheadings.

Variable Length Recording and Segments: The length of audio recordings in general varies a lot, this is especially true for web data. In SLAT, audio recordings are chunked into fixed size segments to train the CNN. The labels for these segments are assumed to be same as label for whole recording. However, in terms of implementation this would be a cumbersome process. It is also a computationally inefficient, the common computations across segments should not be redone.

Hence, to handle variable length recordings efficiently, we propose to have a fully convolutional CNN architecture. In Fig 1 the FCv1 and Fcv2 layers are convolutional layers, unlike most CNN architectures [11, 13, 6, 10] where the last few layers are fully connected dense layers. FCv1 in $\mathcal{N}_W$ consists of 256 filters of receptive field size 4×4. Stride and padding are 1 and 0 respectively and ReLU activation is used. FCv2 is another fully convolutional layer with *C* (number of classes) filters of size 1×1 and Sigmoid Activation. The above changes provides an elegant way to process a variable length recording through the network.

Consider a training recording with 1280 logmel frames i.e $\mathcal{X} \in R^{1280X128}$. If we pass this input through $\mathcal{N}_W$, the output from FCv2 will be $C \times 37 \times 1$ where C is number of classes, 37 is number of segments. This output is same as considering a window (segment) of 128 logmel frames, moving the window by 32 frames to get next window (segment) and forwarding the windows one by one through a corresponding network of fully connected layers at end. We will have 37 segments and we will obtain outputs for each segments one by one. However, FCv1 and Fcv2 allows us to process the whole recording in an efficient manner in one forward pass. Hence, we get output for each segment over all classes in a single forward pass. The exact window (segment) and hop size can be controlled by designing the convolution and pooling layers accordingly. In our case, to keep things simple, we kept a window and hop size of 128 and 32 mel-frames. This was achieved by convolution layers which does not affect the output size along time and frequency and then using 2×2 pooling layers which reduces size by half along both dimensions.

Computing Recording Level Loss: For weakly labeled data labels are available only at recording level. Hence, we do not know the labels for each segments and we need to compute loss at the recording level. To compute the loss for a given recording, we argue that the loss should be computed with respect to the segment which gives the maximum output (for that event class). This is idea is adopted from multiple instance learning methods where "max" instance often characterizes bags [7, 18, 4]. One can potentially also use "average" over all segments.

This is implemented in the network through a global pooling layer. FCv2 produces output for each class for all segments. The recording level score for each class is then obtained by applying a *Global-Max pooling* layer after FCv2. This layer takes max over outputs of all segments to give C X 1 dimensional output for the recording. Once we have that we can compute losses using Eq 1.

Multi-label Training: Often several audio events are simultaneously present in an audio recording. Hence, we design our network for multi-label training and prediction. The sigmoid output in the last layer can be considered as class specific posterior for any given input. Binary cross entropy function as shown in Eq 1 is then used to compute loss with respect to each class.

$$l(y_c, p_c) = -y_c * log(p_c) - (1 - y_c) * log(1 - p_c), \quad L(\mathcal{X}, y) = \frac{1}{C} \sum_{c=1}^{C} l(y_c, p_c) \tag{1}$$

In Eq 1 y_c and $p_c = \mathcal{N}_S(\mathcal{X})$ are target and network output for c^{th} class respectively. The overall loss function is the mean of losses over all classes and is given by $L(\mathcal{X}, y)$.

Prediction: During prediction mel spectra for the test recording at each FFT size are forwarded through the network and then the average output score for each class across all FFT sizes is considered as the final output.

3 Experiments and and Results

3.1 Datasets

Urbansounds (US): Urbansounds [14] dataset gives us human annotated weakly labeled data for 10 sound events. The source of audios in this dataset is Freesound [1]. A total of 1302 recordings, amounting to about 27 hours is present in the dataset, with recording length varying from a few seconds to around 10 minutes. The dataset comes pre-divided into 10 folds. We use the first 4 for training (533 recordings), next 2 (262 recordings) for validation and the last 4 (502 recordings) as testing set.

Event Name	US Training		Youtube Training	
	SLAT	PROP.	SLAT	PROP.
Air Conditioner	0.209	0.424	0.073	0.099
Car Horn	0.432	0.650	0.403	0.696
Children Playing	0.731	0.876	0.532	0.760
Dog Bark	0.855	0.912	0.807	0.816
Drilling	0.687	0.779	0.299	0.296
Engine Idling	0.273	0.416	0.220	0.228
Gunshot	0.831	0.954	0.826	0.851
Jackhammer	0.667	0.721	0.088	0.350
Siren	0.599	0.776	0.557	0.505
Street Music	0.858	0.864	0.663	0.599
MAP	**0.614**	**0.737**	0.447	**0.520**

Event Name	US Training		Youtube Training	
	SLAT	PROP.	SLAT	PROP.
Air Conditioner	0.059	0.074	0.137	0.199
Car Horn	0.114	0.369	0.345	0.564
Children Playing	0.558	0.496	0.279	0.487
Dog Bark	0.501	0.588	0.709	0.714
Drilling	0.395	0.340	0.271	0.401
Engine Idling	0.165	0.173	0.344	0.423
Gunshot	0.509	0.627	0.653	0.864
Jackhammer	0.098	0.150	0.120	0.199
Siren	0.579	0.593	0.768	0.774
MAP	**0.331**	**0.379**	**0.403**	**0.514**

Table 1: Average Precision for sound events. **Left Table:** US Test Set. **Right Table:** Audioset Test

Youtube Training Set: The importance of weakly labeled learning lies in being able to obtain labeled data directly from web without any human labeling effort and be able to train robust AEC models from these data. To show this, we collect training data for the 10 sound events directly from Youtube. We use a simple approach to collect data with weak labels from Youtube. We form the text query for searching on Youtube by adding the keyword "sound" to the event name, e.g *children playing sound*. We then select the top 125 videos under 10 minutes duration retrieved by Youtube and mark them to contain that event. The duration of recording varies from 0.6 seconds to ~10 min., with average duration of 2.1 minutes. The total audio data collected is around 48 hours.

Audioset Test Set: To test our approach on Youtube quality data, we used the "Eval" set from Audioset dataset [5]. The source of Audioset is also Youtube. 9 events out of the 10 events considered are present in Audioset (except *street music*) and we will present results only for 9 events. A total of 761 test recordings exist with durations of 10 seconds for most cases. We ensured that these test recordings are not part of our Youtube training set.

Note that, due to mel spectra feature extraction at 4 different FFT scales, the total experimental data is four times in all cases. Similar to [6] Average Precision (AP) is used to measure performance. Due to space constraints we request readers to the arXiv version [8] of the paper for other details.

Left Table in 1 shows performance for SLAT and proposed method on US test set. For US dataset training, we observed that weakly labeled framework always outperforms corresponding training done with simplistic strong label assumption. In terms of MAP, we see an absolute improvement of $\mathbf{12.3}\%$. For several events, performance improves by over $10 - 15\%$ in absolute terms, showing that weakly labeled data needs to be considered as weakly labeled and strong label assumptions hurts learning. For training with Youtube collected dataset also, we see our method outperforms SLAT by $\mathbf{7.3}\%$ in terms of MAP, justifying weak label learning over SLAT. The quality, nature and form of Youtube training data is very different from Urbansounds dataset. It is much more noisy, overlapping sounds are dominant. Hence, US training set outperforms Youtube training on US test set.

Right Table in 1 shows performance on the Audioset test set. Our weak label CNN is once again superior to SLAT for both training datasets. It outperforms SLAT by an absolute $\mathbf{11.3}\%$ for Youtube training. On this dataset clearly Youtube training data does better. This shows that for content based retrieval of wild multimedia data on web and for applications where the need is to recognize events in consumer generated data, it is important to train on such data. Recognizing classes such as *Air Conditioner* and *Jackhammer* in Audioset is hard. These events can be easily confused with other types of events and noises which are often present in Youtube quality data.

4 Conclusions

In this paper we proposed a deep CNN framework to learn audio event recognition using web data. We showed it is possible to collect weakly labeled data directly from web and then train AEC models using that. Our proposed CNN based learning framework nicely incorporates weakly labeled nature of the audio data. It outperforms training mechanism which makes strong label assumption by a considerable margin. Moreover, the proposed framework can efficiently handle recordings of variable length during training as well as testing. No pre-processing to segment the recording into fixed length is required. Besides a better and smoother implementation process this is also computationally more efficient.

Our proposed framework can perform temporal localization as well, the output of $\mathcal{N}_W$ before GPL layer is the output for different segments. Hence, in principle the network can roughly locate where an event occurred. We plan to investigate this aspect of the framework in future.

References

[1] Freesound website. `https://freesound.org/`.

[2] Pradeep K Atrey, Namunu C Maddage, and Mohan S Kankanhalli. Audio based event detection for multimedia surveillance. In *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, volume 5. IEEE, 2006.

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[4] Daniel R Dooly, Qi Zhang, Sally A Goldman, and Robert A Amar. Multiple instance learning of real valued data. *The Journal of Machine Learning Research*, 3:651–678, 2003.

[5] Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *IEEE ICASSP*, 2017.

[6] Shawn Hershey, Sourish Chaudhuri, Daniel PW Ellis, Jort F Gemmeke, Aren Jansen, R Channing Moore, Manoj Plakal, Devin Platt, Rif A Saurous, Bryan Seybold, et al. Cnn architectures for large-scale audio classification. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, pages 131–135. IEEE, 2017.

[7] Anurag Kumar and Bhiksha Raj. Audio event detection using weakly labeled data. In *24th ACM International Conference on Multimedia*. ACM Multimedia, 2016.

[8] Anurag Kumar and Bhiksha Raj. Deep cnn framework for audio event recognition using weakly labeled web data. *arXiv preprint arXiv:1707.02530*, 2017.

[9] Janvier Maxime, Xavier Alameda-Pineda, Laurent Girin, and Radu Horaud. Sound representation and classification benchmark for domestic robots. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6285–6292. IEEE, 2014.

[10] Huy Phan, Lars Hertel, Marco Maass, and Alfred Mertins. Robust audio event recognition with 1-max pooling convolutional neural networks. *arXiv preprint arXiv:1604.06338*, 2016.

[11] Karol J Piczak. Environmental sound classification with convolutional neural networks. In *Machine Learning for Signal Processing (MLSP), 2015 IEEE 25th International Workshop on*, pages 1–6. IEEE, 2015.

[12] Karol J Piczak. Esc: Dataset for environmental sound classification. In *Proceedings of the 23rd ACM International Conference on Multimedia*, pages 1015–1018. ACM, 2015.

[13] Justin Salamon and Juan Pablo Bello. Deep convolutional neural networks and data augmentation for environmental sound classification. *IEEE Signal Processing Letters*, 24(3):279–283, 2017.

[14] Justin Salamon, Christopher Jacoby, and Juan Pablo Bello. A dataset and taxonomy for urban sound research. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 1041–1044. ACM, 2014.

[15] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. Yfcc100m: The new data in multimedia research. *Communications of the ACM*, 59(2):64–73, 2016.

[16] Wei Tong, Yi Yang, Lu Jiang, Shoou-I Yu, ZhenZhong Lan, Zhigang Ma, Waito Sze, Ehsan Younessian, and Alexander G Hauptmann. E-lamp: integration of innovative ideas for multimedia event detection. *Machine vision and applications*, 25(1):5–15, 2014.

[17] Shoou-I Yu, Lu Jiang, Zexi Mao, Xiaojun Chang, Xingzhong Du, Chuang Gan, Zhenzhong Lan, Zhongwen Xu, Xuanchong Li, Yang Cai, et al. Informedia@ trecvid 2014 med and mer. In *NIST TRECVID Video Retrieval Evaluation Workshop*, volume 24, 2014.

[18] Zhi-Hua Zhou and Min-Ling Zhang. Neural networks for multi-instance learning. In *Proceedings of the International Conference on Intelligent Information Technology, Beijing, China*, pages 455–459, 2002.