LEARNING SOUND EVENTS FROM WEBLY LABELED DATA

Anurag Kumar, Ankit Shah, Alex Hauptmann, Bhiksha Raj
Carnegie Mellon University, Pittsburgh, PA, USA
argxkr@gmail.com,aps1,alex,bhiksha@cs.cmu.edu

ABSTRACT
In the last couple of years, weakly labeled learning for sound events has turned out to be an exciting approach for audio event detection. In this work, we introduce *webly labeled* learning for sound events in which we aim to remove human supervision altogether from the learning process. We first develop a method of obtaining labeled audio data from the web (*albeit noisy*), in which no manual labeling is involved. We then describe deep learning methods to efficiently learn from these webly labeled audio recordings. In our proposed system, *WeblyNet*, two deep neural networks *co-teach* each other to robustly learn from webly labeled data, leading to around 17% relative improvement over the baseline method. The method also involves transfer learning to obtain efficient representations.

Index Terms— Audio Events, Weakly Labeled, Webly Labeled

1. INTRODUCTION

As artificial intelligence becomes an increasingly integral part of our life, it is imperative that automated understanding of sounds too gets integrated into everyday systems and devices. Sound event detection and understanding has a wide range of applications [1], and hence, in the past few years, the field has received considerable attention in broader areas of machine learning and audio processing.

One long-standing problem in the field of audio event detection (AED) has been the availability of labeled data. Labeling sound events in an audio stream requires marking their beginnings and ends. Annotating audio recordings with times of occurrences of events is a laborious and resource intensive task. Weakly-supervised learning for sound events [2] addressed this issue by showing that it is possible to train audio event detectors using *weakly labeled* data: audio recordings, here, are tagged only with presence or absence of the events as opposed to the time stamp annotations in of *strongly labeled* audio data.

Weakly labeled AED has gained significant attention since it was first proposed and has turned out to be the preferred and the most promising approach for scaling AED. Several weakly labeled methods have been proposed [3, 4, 5, 6, 7], to mention a few. Audio events dataset much larger than before have been possible through weak labels [8, 9]. It also features in the annual DCASE challenge for sound events and scenes recognition [1].

Being able to work with weak labels is, however, only half the story. Even weak labeling, when done manually, can become challenging on a large scale; labeling a large number of audio recordings for a large number of classes is non-trivial. Datasets along the lines of Audioset [8] are not easy to create and require considerable resources. However, as pointed out in [2], a major motivation of weakly labeled learning is that it enables us to exploit the massive amount of data on the web *without employing manual annotation*.

The web provides us with a rich resource from which weakly-labeled data could be automatically derived. This can remove the resource intensive processing of creating the training data manually and opens up the possibility of entirely automated training. However, this brings up a new problem – the weak labels associated with these recordings, having been automatically obtained through some means, are likely to be noisy or plain wrong. We call such data ***webly labeled*** and the challenge now extends to being able to learn from such noisy weakly labeled data.

Learning sound events from webly labeled data has received little to no attention. There have been a few works on webly supervised learning for recognizing visual objects and concepts [10, 11, 12]. On the AED front, the main prior work is [13], where webly labeled data have been employed; however, to counter the noise in the labels, [13] adds strongly labeled supporting data in a graph-based approach to provide additional supervision. Needless to say, the strong labels in the support data are manually obtained.

Our objective in this paper is to eliminate human supervision altogether, from the process of learning from webly labeled data.

Challenges in Webly Labeled Learning: Webly labeled learning involves several challenges. The first one is obtaining the webly labeled data itself. Obtaining quality exemplars from the web has been well documented in several computer vision works [10, 14, 15, 12]. This applies for sound events as well and is, in fact, harder due to the complex and intricate ways we describe sounds [1].

The next big challenge is that webly labeled data are always going to contain noisy labels. This inherently makes the training process harder. The third factor which further complicates the learning is the undesirable intra-class variations. Human labeling favor consistency in the acoustic signatures of sounds. Webly labeled data can contain unexpected variability in acoustic characteristics of a sound class which can make it harder to learn the underlying representation and structure of the class.

Moreover, manual annotations keeps in check the overall amount of signal noise in the data. Even if the source of data is the web (e.g Audioset), manual labeling ensures that the sound is at least audible to most human subjects. However, for webly labeled data the signal noise is "unchecked" and the sound event, even if present, might be heavily masked by other sounds or noise. Thus, signal noise in the webly labeled data is difficult to quantify and remains an open research topic for future works.

We develop an entire framework to deal with such webly labeled data. We begin by collecting a webly labeled dataset using a video search engine as the source. We then propose a deep learning based system for effectively learning from this webly labeled data. Our primary idea is that two neural networks can co-teach each other to robustly learn from webly labeled data. The two networks use two different views of the data. Since the labels are noisy, we argue that one cannot rely only on the loss with respect to the labels to train the networks. Instead, the agreement between the networks can be used to address this problem. Hence, we introduce a method to factor the agreement between the networks in the learning process. Our system also includes transfer learning to obtain efficient feature representations.

2. WEBLY LABELED LEARNING OF SOUNDS

2.1. Collecting Training Data

Obtaining training audio recordings is the first step in the learning process and is a considerably hard open problem on its own. The most popular approach in webly supervised systems in vision has

[1] http://dcase.community/

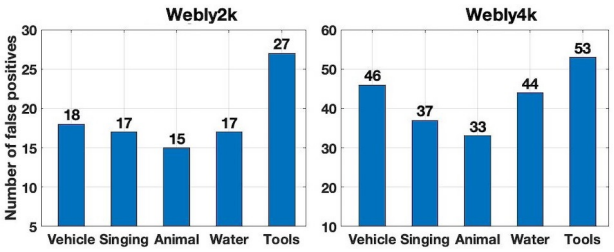

Fig. 1. False Positive counts for the 5 sound classes with highest FP.

been text query based retrieval from search engines [14, 15, 10]. Our approach is along similar lines where we use text queries to retrieve potential exemplars from a video search engine, YouTube.

We must first select a "vocabulary" of sounds – terms used to describe sounds. In this paper, we use a subset of 40 sound events from Audioset, chosen based on several factors. These include preciseness in event names and definitions, the quality of metadata-based retrieval of videos from YouTube, the retention of sound hierarchies, and the number of exemplars in Audioset (larger is better).

Obtaining Webly Labeled Data: Using just the sound name itself as text query on YouTube leads to extremely noisy retrieval. [16] argued that humans often use the phrase "sound of" in texts before referring to a sound. Based on this intuition we augment the search query with the phrase "sound of". This leads to a dramatic improvement in the retrieval of relevant examples. For example, using *"sound of dog"* instead of *"dog"* improves the relevant results (sound event actually present in the recordings) by more than 60% in the top 50 retrieved videos. Hence, for every sound class, we use the phrase "sound of <sound-class>" as the search query for retrieving example recordings of each class. The top K retrieved video (audio) recordings for each class are automatically marked to contain the sound event.

We formed two datasets using the above strategy. The first one referred to as ***Webly-2k*** uses top 50 retrieved videos for each class and has around 1,900 audio recordings. The second one, ***Webly-4k***, uses the top 100 retrieved videos for each class and contains around 3,800 recordings. Note that, some of the recordings are retrieved for multiple classes, and hence, the datasets are multi-labeled, similar to Audioset. Only recordings under 4 minutes are considered.

Analysis of the Dataset: The average duration of the *Webly-2k* set is around 111 seconds resulting in a total of around 60 hours of data. Webly-4k is around 108 hours of audio with an average recording duration of 101 seconds.

As mentioned before, label noise is expected in these datasets. To analyze this, we manually verified the positive exemplars of each class and estimated the number of false positives (FP) for each class. Clearly, the larger *Webly-4k* contains far more noisy labels than *Webly-2k*. Figure 1 shows the FP counts for 5 classes with the highest false positives. Note that, for these classes, 30-50% of the examples are wrongly labeled to contain the sound when it is actually not present. However, FP values can be low also for some classes, e.g., *Piano* and *Crowd*. Another form of label noise is false negatives. Estimating false negatives requires one to manually check all of the recordings for all classes, which makes the task considerably difficult. Even the Audioset dataset has not been assessed for false negatives and contains false negatives. Please visit our paper webpage `https://ankitshah009.github.io/weblynet` for full details on our webly labeled data.

Note that, the manual verification was done only for analysis; the actual goal is to learn from noisy-labeled Webly-4k (or -2k) directly.

2.2. Approach: WeblyNet

One of the most critical challenges of webly labeled learning is *noisy labels*. Robustly training neural networks with noisy labels is still a hard open problem [17]. Several methods have been proposed, especially in the visual domain [18, 10, 11, 19, 19, 17].

Our approach is based on the idea of training two networks together in which they see different views of the same data as in multi-view learning. Multi-view learning methods (e.g., co-training [20][21]) are primarily semi-supervised learning methods where learners are trained on different views of the data, and the goal is to maximize their agreement on the unlabeled data. Our proposed method exploits this central idea of the agreement between classifiers to address the challenges of webly labeled data.

The intuition is as follows: Two (or more) independent classifiers operating on noisily labeled data are likely to agree with the provided label when it is correct. When the given label is *incorrect*, the classifiers are unlikely to agree with it. They are, however, likely to agree with *one another* if both of them independently identify the correct label.

Figure 2 shows the overview of the proposed method. The two networks "Network 1" and "Network 2" take as input two different views of the data. The networks are trained jointly by combining their individual loss functions and a third divergence term which explicitly measures the agreement between the two networks. The individual losses provide supervision from given labels, and the mutual-agreement provides the supervision when the labels are noisy.

The idea of two networks co-teaching each other for noisy labels has been explored in some recent works [22, 23]. [22] gives a "when to update rule" where networks are updated when they disagree. In [23], one network samples instances with small losses and these samples are then used to teach the other network.

Our proposed method is different from these works as it explicitly ties in the co-teaching of the networks by having a disagreement measure in the loss term. Moreover, in our method, the two networks are operating on different views of the data and hence have different learning abilities. Due to this, they will not fall in the degenerate situation where both networks essentially end up learning the same thing. This allows us to combine the classifiers' outputs during the prediction phase which further improves the performance. We refer to our overall system as WeblyNet.

Two Views of the Data: Our primary representation of audio recordings is embeddings obtained from Google's VGGish [24] network. It gives a 128-dimensional embedding for each 1 second of audio. Hence, an audio recording $\mathcal{R}$, in this first view, is represented by a feature matrix $X_1 \in \mathcal{R}^{NX128}$, where N depends on the duration of the audio. The temporal structure of the audio is maintained by stacking the embedding sequentially in X_1. The first network $\mathcal{N}_1$ is trained on these features.

Several methods exist in the literature for generating multiple views from a single view [21]. In this work, we propose to use non-linear transforms through a neural network to generate the second view (X_2) of the data. Moreover, given the noisy nature of webly labeled data, we use the idea of transfer learning to obtain an effective second view of the data [4, 25].

We first train a network ($\mathcal{N}_1$ with C = 527) on the Audioset dataset and then use this trained model to obtain feature representations for our webly labeled data. More specifically, the F2 layer is used to obtain 1024-dimensional representations for the audio recordings by averaging the outputs across all 1-second segments. This representation learning through knowledge transfer can be crucially useful in webly labeled data where a higher level of signal noise and intra-class variation is expected.

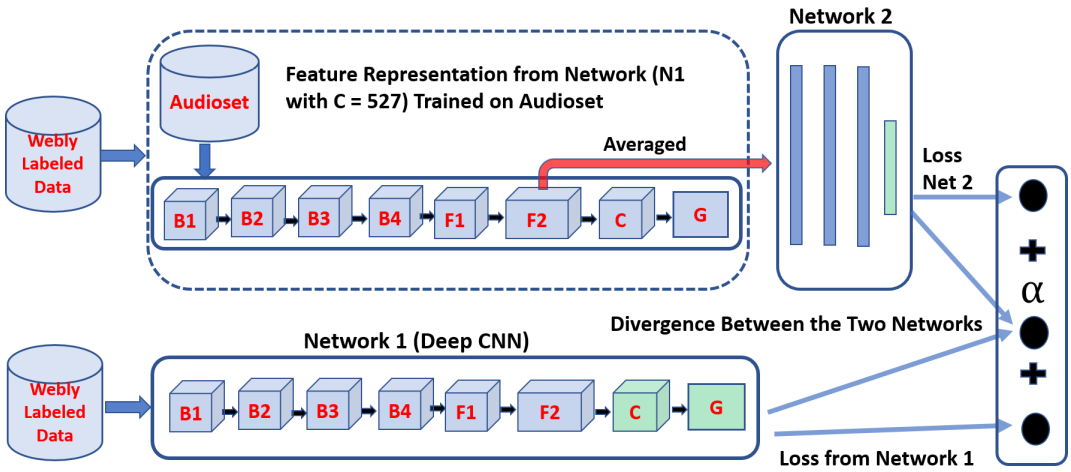

Fig. 2. WeblyNet System: Network 1 ($\mathcal{N}_1$) is a deep CNN with first view of data as input. Network 2 ($\mathcal{N}_2$) takes in the second view of data obtained through transfer learning. The networks are trained together to co-teach each other.

Network 1 ($\mathcal{N}_1$): $\mathcal{N}_1$ is trained on the first (X_1) audio representations. It is a deep CNN. The layer blocks from B1 to B4 consists of two convolutional layers followed by a max-pooling layer. The number of filters in both convolutional layers of these blocks are, { B1:64, B2:128, B3:256, B4:256 }. The convolutional filters are of size 3×3 in all cases, and the convolution operation is done with a stride of 1. Padding of 1 is also applied to inputs of all convolutional layers. The max-pooling in these blocks are done using a window of size 1×2, moving by the same amount. Layer F1 and F2 are again convolutional layers with 1024 filters of size 1×8 and 1024 1024 filters of size 1×1 respectively. All convolutional layers from B1 to F2 consists includes batch-normalization [26] and ReLU [27] activations. The layer represented as C is the segment level output layer. It consists of C filters of size 1×1, where C is the number of classes in the dataset. This layer has a sigmoid activation function. The segment level outputs are pooled through a mapping function in the layer marked as G, to produce the recording level output. In this case, we use the average function to perform this mapping.

The network architecture $\mathcal{N}_1$ is based on the fact that this architecture achieves state-of-the-art results on Audioset. We use the same architecture to train on Audioset, with the only difference being that the number of channels layer C in $\mathcal{N}_1$ (see Fig 2) has C = 527 filters corresponding to 527 classes in the Audioset dataset.

Network 2 ($\mathcal{N}_2$): The network $\mathcal{N}_2$ (with X_2 as inputs) consists of 3 fully connected hidden layers with 2048, 1024 and 1024 neurons respectively. The output layer contains C number of neurons. A dropout of 0.4 is applied after first and second hidden layers. ReLU activation is used in all hidden layers and sigmoid in the output layer.

Training WeblyNet: Given the multi-label nature of datasets, we first compute the loss with respect to each class. The output layer of both $\mathcal{N}_1$ and $\mathcal{N}_2$ gives posterior outputs for each class. We use the binary cross-entropy loss, defined with respect to c^{th} class as, $l(y_c, p_c) = -y_c * log(p_c) - (1 - y_c) * log(1 - p_c)$. Here, y_c and $p_c = \mathcal{N}(X)$ are the target and the network output for c^{th} class, respectively. The overall loss function with respect to the target is the mean of losses over all classes, as shown in Eq 1

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

In the WeblyNet system, $\mathcal{N}_1$ and $\mathcal{N}_2$ co-teach each other through the following loss function

$$\mathcal{L}(X_1, X_2, y) = \mathcal{L}(\mathcal{N}_1(X_1), y) + \mathcal{L}(\mathcal{N}_2(X_2), y) + \alpha \cdot D(\mathcal{N}_1(X_1), \mathcal{N}_2(X_2)) \tag{2}$$

The first two terms in Eq2 are losses for the two networks with respect to the target. $D(\mathcal{N}_1(X_1), \mathcal{N}_2(X_2))$ is the divergence measure between the outputs of the two networks. The α term in Eq 2 is a hyperparameter and controls the weight given to the divergence between the outputs of the two networks in the total loss. This can be set through a grid search and validation.

The divergence, $D(\mathcal{N}_1(X_1), \mathcal{N}_2(X_2))$, between the networks can be measured through a variety of functions. We found that the generalized KL-divergence worked best [28]. Note that, the outputs from the two networks do not sum to 1. The generalized KL divergence is defined as $D_{KL}(\mathbf{x}||\mathbf{y}) = \sum_{i=1}^{d} x_i log(\frac{x_i}{y_i}) - \sum_{i=1}^{d} x_i + \sum_{i=1}^{d} y_i$. $D_{KL}(\mathbf{x}||\mathbf{y})$ is non-symmetric and is not a distance measure. We use $D(\mathbf{x}, \mathbf{y}) = D_{KL}(\mathbf{x}||\mathbf{y}) + D_{KL}(\mathbf{y}||\mathbf{x})$ to measure the divergence between the outputs of the two networks. This measure is symmetric with respect to $\mathbf{x}$ and $\mathbf{y}$. If $\mathbf{o}^{\mathcal{N}_1}$ and $\mathbf{o}^{\mathcal{N}_2}$ are outputs from $\mathcal{N}_1$ and $\mathcal{N}_2$ respectively then $D(\mathcal{N}_1(X_1), \mathcal{N}_2(X_2))$ is

$$D(\mathcal{N}_1(X_1), \mathcal{N}_2(X_2)) = \sum_{i=1}^{C} (\mathbf{o}_i^{\mathcal{N}_1} - \mathbf{o}_i^{\mathcal{N}_2}) \frac{\mathbf{o}_i^{\mathcal{N}_1}}{\mathbf{o}_i^{\mathcal{N}_2}} \tag{3}$$

During the inference stage, the prediction from WeblyNet is the average of the outputs from the two networks $\mathcal{N}_1$ and $\mathcal{N}_2$.

3. EXPERIMENTS AND RESULTS

WeblyNet is trained on the Webly-4k and Webly-2k training sets. Audio recordings are represented through X_1 and X_2 views (Sec. 2.2). To the best of our knowledge, no other work has done an extensive study of webly supervised learning for sound events. The only other relevant previous work, [13], is computationally not scalable to over 100 hours of data we use in this work. Moreover, it also relies on strongly labeled data in the learning process.

All recordings from the *Eval set* of Audioset are used as the test set. The size of this set is around 4500 recordings for the 40 sound events in our vocabulary. A subset of recordings from the *Unbalanced set* of Audioset is used as the validation set.

Furthermore, to compare our webly supervised learning with a manually labeled set, we also create *Audioset-40* training set. *Audioset-40* is obtained from the *balanced set* of the Audioset by taking all recordings corresponding to the 40 sound events in our vocabulary. This turns out to be over 4,600 recordings.

All experiments are done in PyTorch deep learning toolkit. Hyperparameters are tuned using the validation set. The network is trained using Adam optimization method [29]. Similar to previous works [8, 4], we use Average Precision (AP) as the performance metric for each class and then Mean Average Precision (MAP) of all

Method	**MAP**	**Method**	**MAP**
WLAT [4]	21.3	ResNet-SPDA [30]	21.9
ResNet-Attention [7]	22.0	ResNet (mean pooling) [3]	21.8
M&mmnet-MS [3]	22.6	**Ours $\mathcal{N}_1$**	**22.9**

Table 1. MAP of $\mathcal{N}_1$ compared with state-of-the-art on whole Audioset (527 Sound Events, Training: Balanced Set, Test: Eval Set)

Methods	MAP
$\mathcal{N}_1$-Self (Baseline) (4k)	38.7
$\mathcal{N}_2$-Self (4k)	41.4
WeblyNet (4k)	**45.3**

Methods	MAP
$\mathcal{N}_1$-Self (Baseline) (2k)	38.0
$\mathcal{N}_2$-Self (2k)	41.2
WeblyNet (2k)	**44.0**

Methods	MAP	Methods	MAP
$\mathcal{N}_1$-Self (Baseline)	38.7	$\mathcal{N}_1$ (Co-trained)	43.6
$\mathcal{N}_2$-Self	41.4	$\mathcal{N}_2$ (Co-trained)	43.5
$\mathcal{N}_1$-Self and $\mathcal{N}_2$-Self (Averaged)	43.5	WeblyNet	**45.3**

Table 2. **Upper Tables**: Comparison of systems on Webly-4k (L) and Webly-2k (R). **Lower**: $\mathcal{N}_1$ and $\mathcal{N}_2$ co-teach each other in WeblyNet leading to improvement in their individual performances over training them separately. Results shown on Webly-4k. See Sec. 3.1

sound classes is used as the overall metric for comparison. Please visit **`https://ankitshah009.github.io/weblynet`** for webly labeled data, codes, setup and additional analysis.

Full Audioset Performance: We begin by evaluating the performance of the network architecture $\mathcal{N}_1$ (with C=527) on Audioset. The primary motivation behind this analysis is to show that the architecture of $\mathcal{N}_1$ is capable of obtaining state-of-the-art results on a standard well-known weakly labeled dataset. Table 1 shows comparison with state-of-the-art. Our $\mathcal{N}_1$ is able to achieve state-of-art-performance on Audioset. [3] reports a slightly better MAP of 23.2 using an ensemble of M&mmnet-MS. An ensemble can improve our performance as well.

The performance of $\mathcal{N}_1$ on Audioset shows that it can serve as a good base architecture for our WeblyNet system. Hence, it also serves as the baseline method for comparison.

3.1. Evaluation Webly Supervised Learning

We first train $\mathcal{N}_1$ alone on the webly labeled dataset, and this performance ($\mathcal{N}_1$-Self) is taken as the baseline number. We also train $\mathcal{N}_2$ alone on X_2 features ($\mathcal{N}_2$-Self) to assess the significance of transfer learning in improving webly supervised learning.

The upper two tables in Table-2 shows results for different systems on Webly-4k and Webly-2k training sets. We first observe that WeblyNet leads to an absolute improvement of **6.6%** (**17%** relative) over the baseline method on the Webly-4k training set. Another important fact to note is that the X_2 representations learned through transfer learning leads to considerable improvement over the baseline performance; around 7% and 8.5% relative improvements on the Webly-4k and Webly-2k training sets respectively.

The lower table in Table-2 shows how our proposed system in which both networks co-teach each other leads to an improvement in the performance of the individual networks. Once the WeblyNet system has been trained, we consider the output from individual $\mathcal{N}_1$ (or $\mathcal{N}_2$) as the output of the system. These are referred to as $\mathcal{N}_1$ (Co-trained) and $\mathcal{N}_1$ (Co-trained) respectively in the table. We can observe that the performances of both networks are improved by a considerable amount, over 12.7% for $\mathcal{N}_1$ and over 5% for $\mathcal{N}_2$. We see that a simple combination of the two networks ($\mathcal{N}_1$-Self and $\mathcal{N}_2$-Self (Averaged) also leads improved results. Once the networks are co-trained the individual networks improve and the combined system, WeblyNet, leads to 45.3 MAP.

Comparison With Manual Labeling: Table 3 shows comparison of $\mathcal{N}_1$ trained on Audioset-40 with other systems trained on Webly-4k set. A considerable difference between $\mathcal{N}_1$ trained on Audioset-40 and that trained on our Webly-4k exists, around 15.6%. WeblyNet improves the webly supervised learning, however, compared to the training on manually labeled Audioset-40, a difference of 9.0% still exists. Multiple factors are responsible. First, the challenges of webly labeled learning outlined before. Moreover, human labeled data such as the Audioset favors consistency, which will be evident in both training and *evaluation* data used to test the models; which gives an edge to the manually labeled dataset. Such differences in performances between human-supervised data and non-human supervised webly labeled data has been reported before in vision. [10] mentions that human supervision is extremely hard to beat even by using orders of magnitude more web data.

Method	MAP	Method	MAP	Method	MAP
Audioset-40 $\mathcal{N}_1$-Self	54.3	Webly-4k $\mathcal{N}_1$-Self	38.7	Webly-4k WeblyNet	45.3

Table 3. Webly labeled training vs. manual labeling (Audioset-40)

Sounds	Webly-4k		Webly-2k	
	$\mathcal{N}_1$-Self	WeblyNet	$\mathcal{N}_1$-Self	WeblyNet
Vehicle	28.1	46.9	34.8	36.4
Singing	51.3	52.7	47.8	50.5
Animal	29.7	35.2	29.0	31.1
Water	51.8	59.6	50.4	51.9
Tools	31.9	36.5	40.7	40.6
Avg.	**38.6**	**46.2**	**40.5**	**42.1**

Table 4. AP for 5 classes with very high label noise in webly labeled sets.

Some class specific results: Readers are requested to visit the companion webpage to view all details of class-specific results. To draw some interesting points, we analyze the performance of five classes with high label noise (Fig. 1). Table 4 shows the AP for these sounds. For all of these events, the improvements are considerable, e.g., around 67% and 19% relative improvements for Vehicle and Animal sounds, respectively. Interestingly, $\mathcal{N}_1$'s overall performance goes down for these 5 events as we increase the size of the dataset, from Webly-2k to Webly-4k. A considerable drop in performances are seen for Vehicle and Tools sounds whereas only small improvements Animal and Water sounds are seen. Deep learning methods are expected to improve as we increase the amount of training dataset. However, it is clear that the larger Webly-4k contains too many noisy labels which adversely affects the performances in certain cases. The proposed WeblyNet system is able to address this problem. On an average WeblyNet gives 7.6% absolute (20% relative) over the baseline method when trained on Webly-4k set.

Effect of divergence measure: To analyze the role of the divergence measure in system's improvement, we remove it from the loss function in Eq 2 (that is $\alpha = 0$) and train the system using losses with respect to the targets only. The WeblyNet system, in this case, produces a MAP of 43.5, which is not an improvement over the simple combination of the individually trained networks. This shows that the two networks are actually co-teaching each other by measuring the divergence between their outputs.

4. CONCLUSIONS

Human supervision comes at a considerable cost, and hence we need to build methods which rely on human supervision to the least possible extent. In this paper, we presented webly supervised learning of sound events as the solution. We presented a method for mining web data and then a robust deep learning method to learn from webly labeled data. We incorporated multiple ideas into our WeblyNet system to tackle the challenging problem of learning from webly labeled data. We showed that our proposed method in which two networks co-teach each other leads to a considerable improvement in performance. In the future, we need better methods to mine the web as well as more efficient methods to learn from such data.

5. REFERENCES

[1] Anurag Kumar, *Acoustic Intelligence in Machines*, Ph.D. thesis, Carnegie Mellon University, 2018.

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

[3] Szu-Yu Chou, Jyh-Shing Roger Jang, and Yi-Hsuan Yang, "Learning to recognize transient sound events using attentional supervision.," in *IJCAI*, 2018, pp. 3336–3342.

[4] Anurag Kumar, M. Khadkevich, and C. Fugen, "Knowledge transfer from weakly labeled audio using convolutional neural network for sound events and scenes," in *Acoustics, Speech and Signal Processing (ICASSP), 2018 IEEE International Conference on*. IEEE, 2018.

[5] Brian McFee, Justin Salamon, and Juan Pablo Bello, "Adaptive pooling operators for weakly labeled sound event detection," *arXiv preprint arXiv:1804.10070*, 2018.

[6] Xugang Lu, Peng Shen, Sheng Li, Yu Tsao, and Hisashi Kawai, "Temporal attentive pooling for acoustic event detection," *Proc. Interspeech 2018*, pp. 1354–1357, 2018.

[7] Yong Xu, Qiuqiang Kong, Qiang Huang, Wenwu Wang, and Mark D Plumbley, "Attention and localization based on a deep convolutional recurrent model for weakly supervised audio tagging," *arXiv preprint arXiv:1703.06052*, 2017.

[8] J. Gemmeke, D. Ellis, D. Freedman, A. Jansen, W. Lawrence, R Moore, M Plakal, and M Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 776–780.

[9] E. Fonseca, J. Pons, X. Favory, F. Font, D. Bogdanov, A. Ferraro, S. Oramas, A. Porter, and X. Serra, "Freesound datasets: a platform for the creation of open audio datasets," in *Proceedings of International Society for Music Information Retrieval; 2017. p. 486-93.* International Society for Music Information Retrieval (ISMIR), 2017.

[10] Xinlei Chen and Abhinav Gupta, "Webly supervised learning of convolutional networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015.

[11] Junwei Liang, Lu Jiang, Deyu Meng, and Alexander Hauptmann, "Learning to detect concepts from webly-labeled video data," IJCAI, 2016.

[12] Santosh K Divvala, Ali Farhadi, and Carlos Guestrin, "Learning everything about anything: Webly-supervised visual concept learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3270–3277.

[13] Anurag Kumar and Bhiksha Raj, "Audio event and scene recognition: A unified approach using strongly and weakly labeled data," in *Neural Networks (IJCNN), 2017 International Joint Conference on*. IEEE, 2017, pp. 3475–3482.

[14] Rob Fergus, Li Fei-Fei, Pietro Perona, and Andrew Zisserman, "Learning object categories from internet image searches," *Proceedings of the IEEE*, vol. 98, no. 8, pp. 1453–1466, 2010.

[15] Yan Xia, Xudong Cao, Fang Wen, and Jian Sun, "Well begun is half done: Generating high-quality seeds for automatic image dataset construction from web," in *European Conference on Computer Vision*. Springer, 2014, pp. 387–400.

[16] Anurag Kumar, Bhiksha Raj, and Ndapandula Nakashole, "Discovering sound concepts and acoustic relations in text," *submitted IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.

[17] Daiki Tanaka, Daiki Ikami, Toshihiko Yamasaki, and Kiyoharu Aizawa, "Joint optimization framework for learning with noisy labels," *arXiv preprint arXiv:1803.11364*, 2018.

[18] Jacob Goldberger and Ehud Ben-Reuven, "Training deep neural-networks using a noise adaptation layer," 2016.

[19] Scott Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich, "Training deep neural networks on noisy labels with bootstrapping," *arXiv preprint arXiv:1412.6596*, 2014.

[20] Avrim Blum and Tom Mitchell, "Combining labeled and unlabeled data with co-training," in *Proceedings of the eleventh annual conference on Computational learning theory*. ACM, 1998, pp. 92–100.

[21] Shiliang Sun, "A survey of multi-view machine learning," *Neural Computing and Applications*, vol. 23, no. 7-8, pp. 2031–2038, 2013.

[22] Eran Malach and Shai Shalev-Shwartz, "Decoupling" when to update" from" how to update"," in *Advances in Neural Information Processing Systems*, 2017, pp. 960–970.

[23] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama, "Co-teaching: robust training deep neural networks with extremely noisy labels," *arXiv preprint arXiv:1804.06872*, 2018.

[24] S. Hershey, S. Chaudhuri, D. Ellis, J. Gemmeke, A. Jansen, R Moore, M. Plakal, D. Platt, R. Saurous, B Seybold, et al., "Cnn architectures for large-scale audio classification," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 131–135.

[25] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang, "A survey of transfer learning," *Journal of Big Data*, p. 9, 2016.

[26] Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[27] Vinod Nair and Geoffrey E Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

[28] Arindam Banerjee, Srujana Merugu, Inderjit S Dhillon, and Joydeep Ghosh, "Clustering with bregman divergences," *Journal of machine learning research*, vol. 6, no. Oct, 2005.

[29] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[30] H. Zhang, T. Xu, M. Elhoseiny, X. Huang, S. Zhang, A. Elgammal, and D. Metaxas, "Spda-cnn: Unifying semantic part detection and abstraction for fine-grained recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1143–1152.