KNOWLEDGE TRANSFER FROM WEAKLY LABELED AUDIO USING CONVOLUTIONAL NEURAL NETWORK FOR SOUND EVENTS AND SCENES

Anurag Kumar
Carnegie Mellon University, Pittsburgh, PA, USA
alnu@andrew.cmu.edu

Maksim Khadkevich, Christian Fügen
Facebook Inc. Menlo Park, CA, USA
khadkevich,fuegen@fb.com

ABSTRACT

In this work we propose approaches to effectively transfer knowledge from weakly labeled web audio data. We first describe a convolutional neural network (CNN) based framework for sound event detection and classification using weakly labeled audio data. Our model trains efficiently from audios of variable lengths; hence, it is well suited for transfer learning. We then propose methods to learn representations using this model which can be effectively used for solving the target task. We study both transductive and inductive transfer learning tasks, showing the effectiveness of our methods for both domain and task adaptation. We show that the learned representations using the proposed CNN model generalizes well enough to reach *human level accuracy* on ESC-50 sound events dataset and sets *state of art* results on this dataset. We further use them for acoustic scene classification task and once again show that our proposed approaches suit well for this task as well. We also show that our methods are helpful in capturing semantic meanings and relations as well. Moreover, in this process we also set state-of-art results on *Audioset* dataset using *balanced* training set.

Index Terms— Audio Event Classification, Weak Label Learning, Transfer Learning, Learning Representations

1. INTRODUCTION

Sound plays a crucial role in our interaction with the surroundings. Hence, it is critical for the success of artificial intelligence that machines or computers are able to comprehend sounds as humans do. This has led to considerable interests in sound event detection and classification research in recent years. The motivation also comes from immediate applications such as surveillance [1], content based indexing and retrieval of multimedia [2, 3] to name a few.

Sound event detection and classification has been constrained by the availability of large scale datasets. Labeling sound events in an audio recording is an extremely difficult task. Besides this there are also situations where marking beginnings and ends of a sound event in an audio recording is inherently ambiguous and open for interpretation by the annotator [4]. To address these concerns [5] introduced weak labeling approaches for sound event detection. Recently, a large scale weakly labeled dataset for sound events, *Audioset* [6], has been released. Weak label learning for sound events was also included in this year's DCASE2017 challenge[7].

Although weak labeling addresses data availability constraints to a certain extent, creating large datasets along the lines of *Audioset* is still not easy. Even weak labeling, when done manually can be a resource intensive and time consuming process. Moreover, it might just be difficult to collect large amounts of labeled data in any form in certain cases. For examples, there are sound events which are inherently rare. Deep learning methods such as those based on CNNs is not directly useful in such cases. However, as pointed out by Ellis *et. al.* in Future Perspectives [8], one can attempt to address this problem by transferring knowledge from a model trained on a large dataset. Motivation also comes from computer vision where deep CNN models have been successfully used to transfer knowledge from one domain to another as well as from one task to another [9, 10]. This approach, more generally referred to as *transfer learning* [11] remains more or less unexplored in the context of sound events and scenes. Some audio related works in transfer learning are [12, 13, 15]. In another earlier work [14], models are first trained on one set of sound events and then tested on another set to understand the idea of *objectness* in sounds.

Transfer learning in computer vision has been successful because of the availability of large datasets such as *Imagenet*, which provides a reasonable collection of labeled examples for a large number of visual objects. This allows one to train deep models which can learn enough information from the source data to be useful in solving other tasks. For sounds, the primary problem has been lack of such large scale dataset; by *large scale* we imply both, the vocabulary of sound events as well as the overall dataset size. The vocabulary of sound events is important because a learning algorithm needs to see a wide variety of sound events to learn models which might be useful in solving other tasks.

Due to lack of such large dataset, Soundnet [16] proposed to transfer knowledge from visual models for sound event recognition. They use CNN models trained for visual objects and scenes to teach a feature extractor network for audio. However, it remains to be seen how a more direct approach of audio to audio knowledge transfer can be done.

In this paper, we propose methods to effectively transfer knowledge from a CNN based sound event model trained on a large dataset (Audioset). We first train a deep CNN model on *Audioset*, a dataset which provides weakly labeled audio examples from *YouTube* for 527 sound events. Our proposed CNN for weak label learning works efficiently and smoothly with audio recordings of variable length. This makes it well suited for its application transfer learning. Moreover, it also outperforms previous methods [17] and is computationally much more efficient.

We then use the above CNN model in both *transductive* as well as *inductive transfer learning* scenarios [11]. For transductive learning scenario which might also be referred to as domain adaptation, we use the CNN models for sound event classification on ESC-50 dataset [18]. In this case the target task is still sound event classification, but the audio recordings are from a different domain. For the inductive transfer learning which might also be referred to as task adaptation, we perform acoustic scene classification task on DCASE 2016 dataset [19]. In *task adaptation* the target task is different from source task. In both cases, the CNN model serves as the framework to learn representations for audios which can further be used to train a classifier. We propose different methods to adapt the network for the target tasks to obtain discriminative representations.

Moreover, we show that these representations capture higher level semantic information as well. Our method also helps automatically understand the relationship between acoustic scenes and sound events. To the best of our knowledge, this is the first work which extensively explores and proposes methods to transfer knowledge from a CNN based model trained on a large-scale sound event dataset such as Audioset.

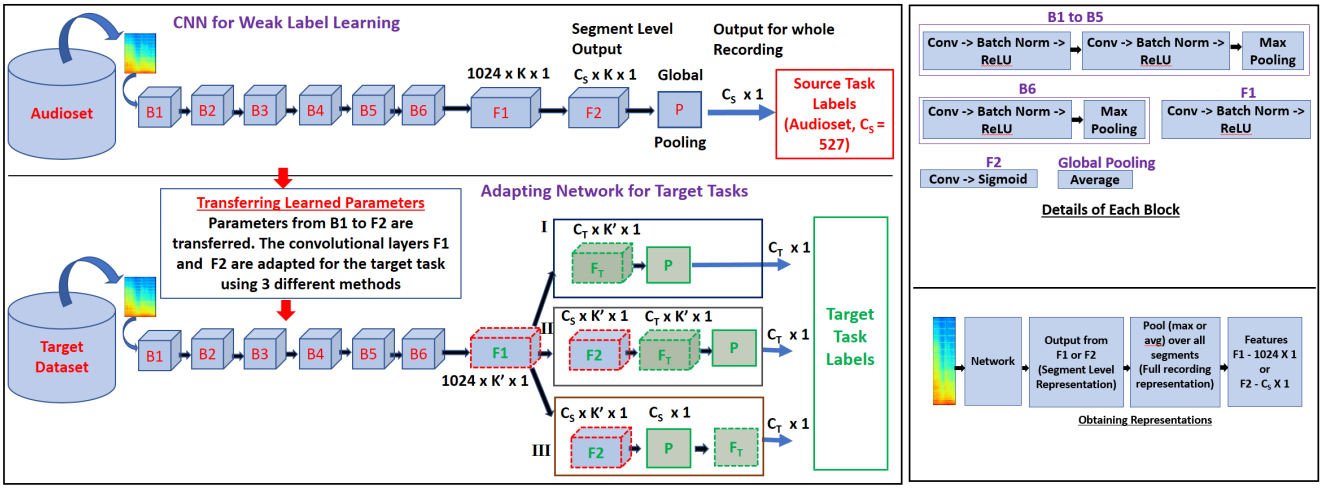


Fig. 1. ***Top Left and Right:*** Deep CNN for Weakly Labeled Audio. B1 to B6 consists of convolutional and pooling layers. F1 and F2 are convolutional layers. P is global pooling layer which transforms segment level output to recording level output. ***Bottom Left:*** Adapting CNN for target task. 3 different methods (I, II, III) are proposed. Parameters from B1 to F1 (or up to F2) are transferred. F1 and (or) F2 onwards are adapted for target task. Newly added layers are shown in green outline. Layers which are updated during task adaptive training are shown in dashed outline. ***Bottom Right***: Obtaining representations for audios. Network can be $\mathcal{N}_S$ or one of $\mathcal{N}_T^I, \mathcal{N}_T^{II}, \mathcal{N}_T^{III}$. See Section 3.2.

2. DEEP CNN FOR WEAKLY LABELED AUDIO

Several CNN approaches have been proposed for sound event classification (SEC), [20, 21] to cite a few. However, most of these works are formulated around *strongly labeled* data. When done on weakly labeled data they are almost always limited in terms of their scale [22, 23, 4]; offering little insight into how well they might generalize in large scale scenarios and be useful for transfer learning. The DCASE 2017 [7] weakly labeled challenge and works based on it also considers only 17 events from Audioset.

[17] analyzes popular CNN architectures such as VGG, Resnet for large scale sound event classification on web videos. However, the training procedure in [17] makes a simplistic strong label assumption for weakly labeled audios. The sound event is assumed to be present in the whole audio recording. For training CNNs, an audio recording is chunked into small segments and fed one by one to the network and the target labels for all segments is set to be same as the label for the whole recording. This training procedure will be referred to as *strong label assumption training* (SLAT).

In this work, we give an alternate approach premised on the ideas proposed in [4], which treats weak labels as weak while training CNNs. Before going into other details, we would like to mention that Logmel spectrograms are used for training CNNs in this work. All audio recordings are sampled at 44.1 KHz sampling frequency. 128 mel bands are used. A window size of 23 ms and an overlap of 11.5 ms is used for obtainining mel features.

2.1. Network Architecture

Our proposed deep CNN framework for weakly labeled audio is shown at the top left and right panels in Figure 1. Block B1 to B5 consists of two convolutional layers (with batch normalization) followed by a max pooling. B6 consists of one colvolutional layer, followed by a max pooling layer. ReLU $(max(0, x))$[24] activation is used in all cases. For convolutional layers in all six blocks, 3×3 filters are used. Stride and padding are fixed to 1. The number of filters used in convolutional layer(s) of blocks B1 to B6 are, $\{B1 : 16, B2 : 32, B3 : 64, B4 : 128, B5 : 256, B6 : 512\}$. Max pooling are done over a 2×2 window, with a stride of 2 by 2.

F1 is also a convolutional layer with ReLU activation. 1024 filters of size 2×2 are used with a stride of 1. No padding is used in F1. F2 is the secondary output layer, a covolutional layer of C_S filters of size 1×1 and sigmoid output. This layer produces segment level output ($C_s \times K \times 1$), where C_s is the number of classes (in source task) and K is the number of segments. The segment level outputs are aggregated using a *global pooling* layer to produce $C_s \times 1$ dimensional output for the whole recording.

The network scans through the whole input (Logmels) and produces outputs corresponding to segments of 128 logmel-frames moving by 64 frames. For example, an input logmel spectrogram consisting of 896 logmel-frames, that is $X \in R^{896 \times 128}$ (128 mel-bands as stated before), will produce K=13 segments at $F1$ and $F2$. Since in weakly labeled audio we have labels for the full recording, the outputs at segment level are pooled to obtain the full recording level output. The loss is then computed with respect to this recording level output. Hence, this network ($\mathcal{N}_S$) treats weak labels as weak. Overall, it is a VGG style [25] CNN for weak label learning of sounds. The segment size and segment hop size can be controlled by the network design. For $\mathcal{N}_S$, these are 128 ($\sim$ 1.5 s) and 64 ($\sim$ 0.75 s) frames respectively in logmel spectorgrams. Note that, if needed segment level outputs can give temporal localization of events [2].

Unlike SLAT where fully connected dense layers are used in CNN architectures, $\mathcal{N}_S$ network is fully convolutional which allows it to process audio recordings of variable length. This makes it well suited for transfer learning.

2.2. Multi-Label Training Loss

For web data, including Audioset, it is expected that multiple sound events might be simultaneously present in the same recording. Hence, we employ a multi-label training loss. The sigmoid output gives class specific posteriors for any given input. The binary cross entropy loss with respect to each class is given by $l(y_c, p_c) = -y_c * log(p_c) - (1 - y_c) * log(1 - p_c)$. y_c and $p_c = \mathcal{N}_S(\mathcal{X})$ are target and network output for c^{th} class respectively. The training loss is the mean of losses over all classes, $L(\mathcal{X}, y) = \frac{1}{C} \sum_{c=1}^{C} l(y_c, p_c)$.

3. TRANSFER AND REPRESENTATION LEARNING

The network $\mathcal{N}_S$ trained on source task audios is used to obtain representations for audios in the target task. The flow of obtaining representations is shown in the bottom right panel of Fig 1. Segment level

outputs from F1 ($1024 \times K \times 1$) and F2 ($C_S \times K \times 1$) serves as base representations for audios. These segments level representations are then mapped to full recording level representations. We apply either $max()$ or $avg()$ for this mapping. Finally, we obtain 1024 (F1) or C_S (F2) dimensional representations for full recordings.

During $\mathcal{N}_S$ training, the blocks from B1 to B6 embeds knowledge from source audio data into $F1$, which is then mapped to source labels by filters in $F2$. This makes F1 well suited for transfer learning, where it can be used to train classifiers for target task. Moreover, outputs from $F2$ gives us a distribution over the source labels, which itself can be useful for the target task when $\mathcal{N}_S$ is trained over a large collection of sound events. We propose two broad methods for representation learning for audios in target tasks using $\mathcal{N}_S$.

3.1. Direct Off-the-shelf Representations

In this method, $\mathcal{N}_S$ is treated in a ready to use mode for obtaining representations. Logmel spectrograms of audio recordings from target task are fed into $\mathcal{N}_S$ and the outputs from F1 and F2 are aggregated over all segments (as described before) to obtain 1024 and C_S dimensional representations respectively.

3.2. Transfer and Adapt for Learning Representations

In the second method, we first adapt the network to target task to extract features which we expect will be more discriminative and better suited for the target classification task. We propose 3 methods to achieve this goal. The methods are shown in Figure 1. In all three methods, the parameters from B1 to B6 are transferred and are not updated during the target adaptation training. Let C_T be the number of classes in the target dataset.

Method I ($\mathcal{N}_\mathbf{T}^\mathbf{I}$) : $\mathcal{N}_T^I$ performs a direct adaptation of F1 to the target task. Here, F2 is replaced by a new covolutional layer (F_T) of C_T filters. Parameters in F1 and F_T are then updated using the training set of the target task. We will call this network $\mathcal{N}_T^I$.

Method II ($\mathcal{N}_\mathbf{T}^\mathbf{II}$) : In $\mathcal{N}_T^{II}$, a new convolutional layer (F_T) of with C_T filters is added after F2 in $\mathcal{N}_S$. This new network, $\mathcal{N}_T^{II}$, is then adapted for the target task. As shown by dashed boundaries in Fig 1, during the adaptive training only F1, F2 and F_T are updated. The idea is to capture target specific information by first transitioning to source label space (F2) and from there going to target label space.

Method III ($\mathcal{N}_\mathbf{T}^\mathbf{III}$) : In $\mathcal{N}_T^{III}$, a new fully connected layer F_T of size C_T is added after the global pooling layer in $\mathcal{N}_S$. Once again, only F1, F2 and F_T are updated during network adaptation training. The motivation behind this network is same as $\mathcal{N}_T^{II}$, except that it tries to learn the mapping at full recording level instead of segment level. Note that in both $\mathcal{N}_T^{II}$ and $\mathcal{N}_T^{III}$, the activation function in F2 is changed to ReLU from sigmoid in $\mathcal{N}_S$.

For all three adapted networks $\mathcal{N}_T^I$, $\mathcal{N}_T^{II}$ and $\mathcal{N}_T^{III}$, if the target task is a multi-label problem, then the activation in final layer is kept as sigmoid and loss function similar to that defined in Section 2.2 is used. However, if the target task audios have single label, then we can use *softmax* output with categorical cross entropy loss.

A few things worth noting. First, the target task can have audio recordings of different length and our proposed methods can handle such cases efficiently. Moreover, the target task dataset can either be strongly or weakly labeled and the proposed methods can be used to learn representations in both cases. Lastly, to emphasize again, the focus is on exploiting $\mathcal{N}_S$ to learn representations for audios in the target task. Classifier such as SVM can be trained on these representations, even if the target task dataset is small.

4. EXPERIMENTS AND RESULTS

We start by showing performance for sound event classification on *Audioset*[1]. We work with all 527 sound events in Audioset for which weakly labeled data is currently available. We compare performance of our $\mathcal{N}_S$ with SLAT ($\mathcal{N}_S^{slat}$). $\mathcal{N}_S^{slat}$ is similar to $\mathcal{N}_S$ except that F1 and F2 are now fully connected layers of size 1024 and C_S. Training is done with fixed size segments of 128 logmel frames as inputs, segments overlap by 64 frames. Loss is computed for each input segment by using recording level labels.

MAUC		MAP	
$\mathcal{N}_\mathbf{S}^\mathbf{slat}$	$\mathcal{N}_\mathbf{S}$	$\mathcal{N}_\mathbf{S}^\mathbf{slat}$	$\mathcal{N}_\mathbf{S}$
0.915	0.927 (**+1.3%**)	0.167	0.213 (**+27.5%**)

Train Time		Inference Time	
$\mathcal{N}_\mathbf{S}^\mathbf{slat}$	$\mathcal{N}_\mathbf{S}$	$\mathcal{N}_\mathbf{S}^\mathbf{slat}$	$\mathcal{N}_\mathbf{S}$
1.0	0.61 (-39%)	1.0	0.67 (-33%)

Lowest 10			Highest 10		
Event	$\mathcal{N}_S^{slat}$	$\mathcal{N}_S$	Event	$\mathcal{N}_S^{slat}$	$\mathcal{N}_S$
Scrape	0.0058	0.0092	Music	0.728	0.749
Crackle	0.0078	0.0097	Siren (Civil Defense)	0.671	0.641
Man Speaking	0.0080	0.0202	Bagpipes	0.646	0.786
Mouse	0.0092	0.0368	Speech	0.631	0.661
Buzz	0.0095	0.0077	Purr (Cats)	0.575	0.600
Squish	0.0102	0.0122	BattleCry	0.575	0.651
Gurgling	0.0111	0.0125	Heartbeat	0.559	0.569
Door	0.0115	0.0685	Harpsichord	0.544	0.630
Noise	0.0116	0.0107	Ringing (Campanology)	0.538	0.690
Zipper	0.0121	0.0161	Timpani	0.538	0.528
Mean	**0.0097**	**0.0203**	**Mean**	**0.600**	**0.651**

Table 1. ***Top Left***: Comparison of MAUC and MAP over all 527 events in Audioset. ***Top Right***: Comparison of Average Relative Training (1 Epoch) and Inference (per test instance) times. ***Bottom Table***: AP comparison for 10 sound events with lowest and highest APs using baseline $\mathcal{N}_S^{slat}$. Section 4.1 and here[3] for details.

All experiments are done in Pytorch[2]. Adam optimization [26] is used for training networks. Validation set is used for tuning parameters and selecting the best model.

We then show experimental results for transfer learning using $\mathcal{N}_S$. For the task adaptive training of $\mathcal{N}_S$, the training set of the target task is used. Learning rate during this process is fixed to 0.0002 and updates are done for 50 epochs, after which the network is used to obtain representations. Linear SVMs [27] are then trained on the representations obtained from different methods. The slack parameter C in SVMs is tuned by cross validation on the training set.

Due to *space constraints* readers are requested to visit **this webpage** [2] for more detailed results and analysis.

4.1. Audioset Results

Audioset[1] dataset consists of weak labels for 527 sound events on YouTube videos. Total dataset consists of over 2 million audio recordings. We use the *balanced training* set for training $\mathcal{N}_S$. *Balanced* training set provides a total of around $22,000$ training audio recordings, with at least 59 examples per class. However, due to multi-label nature of the data the actual number of examples for several classes is much higher. A small subset from *Unbalanced* set of Audioset is used as the validation set in experiments. Results are reported on the full *Eval* set of Audioset, which has around $20,000$ test audio recordings, again with at least 59 examples per class. Similar to [17], Area under ROC curves (AUC) [28] and Average Precision (AP) [29] for each class are used as evaluation metrics.

Table 1 shows Mean AUC (MAUC) and Mean AP (MAP) over all 527 classes in Audioset. An absolute improvement of 1.2 (1.3% relative) in MAUC and 4.6 (27.5% relative) in MAP is obtained using $\mathcal{N}_S$. The top right table shows relative computational times, normalized for comparison. $\mathcal{N}_S^{slat}$ is 33% faster on an average during inference. Hence, more suitable for real applications. During training, on an average it is 39% faster for 1 full pass over training data.

Performance of all classes are available here[2]. Bottom table in Tables 1 shows comparison for 10 sound events for which $\mathcal{N}_S^{slat}$ achieved least and highest APs. For low performance classes, on an average $\mathcal{N}_S$ doubles the AP (**0.0097** to **0.0203**). For easier sound classes **8.5%** relative improvement is obtained using $\mathcal{N}_S$.

[1] https://research.google.com/audioset/, [2] http://pytorch.org/

[2] http://www.cs.cmu.edu/%7Ealnu/TLWeak.htm

Methods	Mean Accuracy
Piczak [20]	64.5 %
Tokozume [30]	71.0 %
Aytar [16]	74.2 %
Proposed (F1)	**83.5** %

Network	F1		F2	
	$max()$	$avg()$	$max()$	$avg()$
$\mathcal{N}_{\mathbf{S}}$	**82.8**	81.6	65.5	64.8
$\mathcal{N}_{\mathbf{T}}^{\mathbf{I}}$	**83.5**	81.3	–	–
$\mathcal{N}_{\mathbf{T}}^{\mathbf{II}}$	**83.5**	81.8	81.9	81.5
$\mathcal{N}_{\mathbf{T}}^{\mathbf{III}}$	83.3	82.6	82.6	81.9

Table 2. ***Left***: ESC-50 Accuracy comparison with baselines. ***Right***: Accuracy comparison of different representations.

Scene	Baseline	$\mathcal{N}_S^{III}$ (F1, $max()$)	Scene	Baseline	$\mathcal{N}_S^{III}$ (F1, $max()$)
Beach	69.3	71.9	Library	50.4	73.6
Bus	79.6	82.4	Metro Station	94.7	80.2
Cafe	83.2	73.8	Office	98.6	85.1
Car	87.2	89.9	Park	13.9	46.9
City Center	85.5	93.3	Residential Area	77.7	63.9
Forest Path	81.0	97.4	Train	33.6	52.3
Grocery Store	65.0	84.6	Tram	85.4	84.0
Home	82.1	69.4	**Mean**	**72.5**	**76.6**

Network	F1		F2		Network	F1		F2	
	$max()$	$avg()$	$max()$	$avg()$		$max()$	$avg()$	$max()$	$avg()$
$\mathcal{N}_{\mathbf{S}}$	72.2	69.8	59.1	60.4	$\mathcal{N}_{\mathbf{T}}^{\mathbf{II}}$	75.5	73.0	73.8	73.9
$\mathcal{N}_{\mathbf{T}}^{\mathbf{I}}$	75.2	73.7	–	–	$\mathcal{N}_{\mathbf{T}}^{\mathbf{III}}$	76.6	73.7	72.5	73.3

Table 3. ***Upper***: DCASE 2016 accuracy comparison with baseline ***Lower***: Accuracy comparison of different representations.

4.2. Domain Adaptation: Sound Event Classification

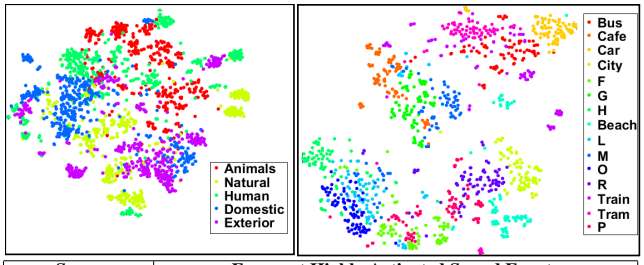
In this section, $\mathcal{N}_S$ is used for learning representations for ESC-50 [18] dataset. ESC-50 dataset consists of a total of 50 sound events, 10 from each 5 broad categories, *Animals* (e.g Dog), *Natural Soundscapes and Water Sounds* (e.g Chirping Birds), *Human Non Speech Sounds* (e.g Clapping), *Domestic Sounds* (e.g. Clock Alarm) and *Exterior Sounds* (e.g Helicopter). The dataset consists of a total of $2,000$ recordings. It comes pre-divided into 5 folds. Four folds are used for training and validation and the remaining fold is used for testing. This is done all 5 ways and average accuracy across all 5 runs accuracy is reported. Human accuracy on this dataset is 81.3%.

Left Table in Tab. 2 compares mean accuracy over all 50 classes with state-of-art on ESC-50 dataset. We outperform the best method by $\mathbf{9.3}\%$, setting state-of-art results on ESC-50. Right table in Tab. 2 shows performance of different representations proposed in this work. Note that, even off-the-shelf F1 representations using $\mathcal{N}_S$ is able to achieve better than human performance on this dataset. This shows that $\mathcal{N}_S$ does an excellent job in capturing sound event knowledge. Task adaptive training gives further improvement. $max()$ mapping for converting segment level representations of F1 or F2 to full recording representations performs better. The sigmoid output from F2 in $\mathcal{N}_S$ does not give good performance using linear SVMs. However, after task adaptive training in $\mathcal{N}_T^{II}$ and $\mathcal{N}_T^{III}$, where F2's activations are changed to ReLU, we obtain good performance from F2 representations. Classwise confusion matrix can be found here[2].

4.3. Task Adaptation: Acoustic Scene Classification

Scenes such as *Park* or *Home* possess complex acoustic characteristics. Often, they are themselves composed of several sound events meshed together in a complex manner. We study the utility of transferring learning for acoustic scenes on DCASE 2016 [19] dataset. It provides 30 seconds examples for 15 acoustic scenes listed in upper table in Tab. 3. The total duration of data is around 9.75 hours. The dataset comes pre-divided into 4 folds, 3 are used for training and remaining for test. This is done all 4 ways are average accuracies across all 4 runs are reported.

Upper table in Table 3 compares accuracies for different acoustic scenes between baseline and one of our proposed method. An absolute improvement of 4.1% over all 15 scenes is observed. For certain scenes which are hard to classify such as *Park* and *Train*, an absolute improvement of 33.0% and 18.7% respectively is obtained. Note that for this task, representations from task adapted networks perform much better compared to those obtained directly from $\mathcal{N}_S$.



Scene	Frequent Highly Activated Sound Events
Cafe	Speech, Chuckle-Chortle, Snicker, Dishes, Television
City Center	Applause, Siren, Emergency Vehicle, Ambulance
Forest Path	Stream, Boat Water Vehicle, Squish, Clatter, Noise, Pour
Grocery Store	Shuffle, Singing, Speech, Music, Siren
Home	Speech, Finger Snapping, Scratch, Dishes, Baby Cry, Cutlery
Beach	Pour, Stream, Applause, Splash - Splatter, Gush
Library	Finger Snapping, Speech, Fart, Snort
Metro Station	Speech, Squish, Singing, Siren, Music
Office	Finger Snapping, Snort, Cutlery, Speech, Cutlery
Residential Area	Applause, Crow, Clatter, Siren
Park	Bird Song, Crow, Stream, Wind Noise, Stream

Fig. 2. ***Top Left***: t-SNE visualizations for ESC-50 ($\mathcal{N}_S$, F1, $max()$). Color coded for 5 higher semantic categories ***Top Right***: t-SNE visualizations for DCASE 2016. First alphabet for some, e.g (F)orest. ***Bottom Table***: Sound Events which are frequently among Top 5 maximally active events for a given scene. Network is $\mathcal{N}_T^{III}$.

$\mathcal{N}_T^{III}$ gives best results, followed closely by $\mathcal{N}_T^{I}$ and $\mathcal{N}_T^{III}$. Once again $max()$ mapping performs better compared to $avg()$.

4.4. Semantic Understanding

We now try to draw some semantic inferences from the proposed methods. Left panel in Fig 2 shows 2 dimensional t-SNE [31] embeddings for representations obtained for ESC-50. The embeddings are color coded for the 5 broad categories in ESC-50 dataset, semantically higher level groups for sound events. One can note from the plot that these representations are capable of capturing higher level semantic information. *Vacuum Cleaner* in *Domestic* closely resembles *Chainsaw*, *Engine* and *Handsaw* in *Exterior* category and its representations also lies closer to *Exterior* sounds (blue dots among purple). Similarly, visualization for 15 acoustic scenes is shown in right panel in Fig. 2.

Acoustic scenes can often be understood through sound events. Each neuron in F2 is essentially representing a sound event class and the activations of these neurons can be used to understand scene-event relations. For each input of a given scene we list the Top 5 maximally activated neurons (events) in F2 (for each segment). We then note the events which occurred most frequently (among top 10) in these lists. These highly active events for some of the scenes are shown in table in Fig 2. We observe that several of these sound events are expected to occur in the corresponding scene. Hence, these scene-events relations are semantically meaningful. This shows that our network managed to successfully transfer knowledge and learn relationships. More analysis on semantics here[2].

5. CONCLUSIONS

We first proposed a CNN based model for weakly labeled learning of sound events. Our model not only sets state-of-art results on Audioset but is also computationally efficient and well suited for transfer learning. We then proposed methods to learn representations for audios in the target task using this CNN model. We set state of art results on ESC-50 dataset, achieving an accuracy of $\mathbf{83.5}\%$, which surpasses human accuracy on this dataset. Besides achieving excellent performance, these methods to transfer knowledge are also helpful in higher level semantic understanding. For example, automatically discovering relationships between scenes and sound events is also an important contribution of this work.

6. REFERENCES

[1] 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*. IEEE, 2006, vol. 5.

[2] 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*, vol. 25, no. 1, pp. 5–15, 2014.

[3] 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*, 2014, vol. 24.

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

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

[6] 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.

[7] A. Mesaros, T. Heittola, A. Diment, B. Elizalde, A. Shah, E. Vincent, B. Raj, and T. Virtanen, "Dcase 2017 challenge setup: Tasks, datasets and baseline system," .

[8] D. Ellis, T. Virtanen, M. Plumbley, and B. Raj, "Future perspective," in *Computational Analysis of Sound Scenes and Events*, pp. 401–415. Springer, 2018.

[9] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1717–1724.

[10] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?," in *Advances in neural information processing systems*, 2014, pp. 3320–3328.

[11] Sinno Jialin Pan and Qiang Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[12] J. Deng, Z. Zhang, E. Marchi, and B. Schuller, "Sparse autoencoder-based feature transfer learning for speech emotion recognition," in *Affective Computing and Intelligent Interaction (ACII), 2013 Humaine Association Conference on*. IEEE, 2013, pp. 511–516.

[13] E. Coutinho, J. Deng, and B. Schuller, "Transfer learning emotion manifestation across music and speech," in *Neural Networks (IJCNN), 2014 International Joint Conference on*. IEEE, 2014, pp. 3592–3598.

[14] A. Kumar, R. Singh, and B. Raj, "Detecting sound objects in audio recordings," in *Signal Processing Conference (EUSIPCO), 2014 Proceedings of the 22nd European*. IEEE, 2014, pp. 905–909.

[15] Aleksandr Diment and Tuomas Virtanen, "Transfer learning of weakly labelled audio," .

[16] Y. Aytar, C. Vondrick, and A. Torralba, "Soundnet: Learning sound representations from unlabeled video," in *Advances in Neural Information Processing Systems*, 2016, pp. 892–900.

[17] 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*. IEEE, 2017, pp. 131–135.

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

[19] Annamaria Mesaros, Toni Heittola, and Tuomas Virtanen, "Tut database for acoustic scene classification and sound event detection," in *24th European Signal Processing Conference*, 2016, vol. 2016.

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

[21] Justin Salamon and Juan Pablo Bello, "Deep convolutional neural networks and data augmentation for environmental sound classification," *IEEE Signal Processing Letters*, vol. 24, no. 3, pp. 279–283, 2017.

[22] Y. Xu, Q. Kong, Qiang Huang, W. Wang, and M. Plumbley, "Attention and localization based on a deep convolutional recurrent model forweakly supervised audio tagging," *Proceedings of Interspeech 2017*, 2017.

[23] Ting-Wei Su, Jen-Yu Liu, and Yi-Hsuan Yang, "Weakly-supervised audio event detection using event-specific gaussian filters and fully convolutional networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 791–795.

[24] 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.

[25] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

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

[27] Rong-En Fan, K Chang, C Hsieh, X Wang, and C Lin, "Liblinear: A library for large linear classification," *The Journal of Machine Learning Research*, 2008.

[28] Tom Fawcett, "Roc graphs: Notes and practical considerations for researchers," *Machine learning*, vol. 31, no. 1, pp. 1–38, 2004.

[29] C. Buckley and E. Voorhees, "Retrieval evaluation with incomplete information," in *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2004, pp. 25–32.

[30] Y. Tokozume and T. Harada, "Learning environmental sounds with end-to-end convolutional neural network," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 2721–2725.

[31] L. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2579–2605, 2008.