The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

Listening to the World Improves Speech Command Recognition

Brian McMahan, Delip Rao*
(brian|delip)@r7.ai
R7 Speech Sciences Inc
San Francisco CA USA

Abstract

We study transfer learning in convolutional network architectures applied to the task of recognizing audio, such as environmental sound events and speech commands. Our key finding is that not only is it possible to transfer representations from an unrelated task like environmental sound classification to a voice-focused task like speech command recognition, but also that doing so improves accuracies significantly. We also investigate the effect of increased model capacity for transfer learning audio, by first validating known results from the field of Computer Vision of achieving better accuracies with increasingly deeper networks on two audio datasets: UrbanSound8k and Google Speech Commands. Then we propose a simple multiscale input representation using dilated convolutions and show that it is able to aggregate larger contexts and increase classification performance. Further, the models trained using a combination of transfer learning and multiscale input representations need only 50% of the training data to achieve similar accuracies as a freshly trained model with 100% of the training data. Finally, we demonstrate a positive interaction effect for the multiscale input and transfer learning, making a case for the joint application of the two techniques.

Introduction

Detection of everyday sounds, such as sounds originating from machinery, traffic sounds, animal sounds, and music is essential for building autonomous agents responsive to their surroundings. This has myriad applications ranging from autonomous vehicles (Chu et al. 2006) to surveillance (Ntalampiras, Potamitis, and Fakotakis 2009) to monitoring noise pollution in cities (Maijala et al. 2018; Salamon, Jacoby, and Bello 2014). Similarly, Spoken Term Recognition (Miller et al. 2007) has broad applications from conversational agents (Sainath and Parada 2015) to monitoring news (Parlak and Saraclar 2008). While many approaches have focused individually on the classification of everyday sounds or recognizing speech, there has been limited investigation into the relationship between models trained on both tasks.

In this paper, we present a systematic study of modern convolutional neural network architectures and transfer learning between two unrelated audio domains – environmental sounds and speech commands. Additionally, we introduce a method for increasing the input resolution of the networks using a single layer of dilated convolutions at multiple scales. Our experiments are designed to answer questions about model capacity, the effect of multiscale dilated convolutions, and the quality of feature learning on audio spectrograms.

In our first experiment, we study model capacity of several convolutional network architectures by measuring the performance at varying depths, with and without multiscale dilated convolutions as inputs on an environmental sound classification task.

Informed by the first experiment, we selected a single convolutional network architecture in our second experiment to evaluate the effectiveness of transfer learning from environmental sounds to speech commands. Models that were pre-trained on environmental sounds and adapted to speech commands were compared to models trained solely on speech commands. Additionally, we investigated whether or not multiscale input through dilated convolutions had a significant impact on the transfer learning. The results of this experiment strongly suggest that the pre-trained convolutional networks with the multiscale inputs are learning important properties about audio spectrograms.

Finally, in our third experiment, we repeated the second experiment of transfer learning from environmental sounds to speech commands but varied the amount of speech command data used for adapting and training. We observe that the pre-trained models need far less data to adapt to the new domain and achieved the much higher accuracy than the models trained only on the speech command data. We also report experiments demonstrating that this gain in accuracy and reduction in training data is additive when multi-scale input representations are used with pre-training.

We conclude by discussing the implications and scope of our experiments.

Related Work

Classifying audio signals has a long and diverse history. In particular, the classification of environmental sounds has attracted researchers from speech to signal processing to bioacoustics employing a range of approaches, such as Support Vector Machines (Temko et al. 2006), Random

*Corresponding author

	No. Audio Samples	Total Hours	No. Classes
UrbanSound8K	8732	8.75	10
Google Speech Commands	64721	17.97	30

Table 1: Descriptions of the two labeled audio datasets that were used, UrbanSound8K and Google Speech Commands, in terms of the number of audio samples, total hours, and total number of classes.

Forests Classifiers (Piczak 2015b), and Multi Layer Perceptrons (Inkyu Choi and Kim 2016). Recently, (Piczak 2015a) and (Salamon and Bello 2016) show Convolutional Neural Networks outperform the traditional methods. Despite the success, neither of these approaches have investigated extremely deep networks (100+ layers) on audio data, one of the goals of this paper.

Relatedly, automatic tagging of music has seen several convolutional networks (Dieleman and Schrauwen 2014; Choi, Fazekas, and Sandler 2016; Lee and Nam 2017), but the networks have been relatively small compared the ones we investigate in this paper. In contrast, domains of audio classification have not seen the systematic application of the increasingly deeper convolutional network architectures that have immensely advanced Computer Vision (Deng et al. 2009; He et al. 2016; Huang et al. 2016).

For audio classification, however, only recently did Hershey et al. (2017) apply a 50-layer Residual Network (also called ResNets) (He et al. 2016) and a 48-layer Inception-V3 (Szegedy et al. 2015) network to classify the soundtracks of videos. We extend the audio classification task to models larger than a 100 layers. Our largest network is a 169 layers deep that we were able to train on a single NVIDIA Titan X GPU in 20 minutes on the UrbanSound8K dataset (8 hours of training data), without needing any specialized large-scale training infrastructure.

Incorporating information from multiple scales is a challenge to convolutional networks, but recently dilated convolutions have shown efficacy in doing so for image classification tasks (Yu and Koltun 2015). Dilations were successfully used by Oord et al.(2016) for a text-to-speech task where the dilated convolution layers are applied hierarchically as a generative model of audio waveforms. Previous works on using multiscale spectrogram (Dieleman and Schrauwen 2014; Choi, Fazekas, and Sandler 2016; Lee and Nam 2017) do not study the effect of multiscale convolutions on spectrogram features. Perhaps the closest work to ours in integrating multiple scales of information in a single layer is Pons et al. (2017), but this work focuses on uses convolutions of heuristically-derived sizes, while we leverage the computational efficiency of dilated convolutions in a simple and straight forward utilization. To the best of our knowledge, this is the first work to systematically study the effect of multiple scales of dilated convolutions for audio classification.

A prominent use of convolutional neural networks in Computer Vision is to utilize transfer learning to classify new image categories (Zeiler and Fergus 2014). Xu et al. (2014) demonstrate transferring representations from speech models trained in one language to another (English to Chinese). Choi et al. (2017) and Hamel et al. (2013) demonstrate transfer learning between one music task and another music task. Wang and Metze (2017) show positive results of transferring latent representations learned from a large unlabeled sound event corpus to a smaller labeled sound event corpus.

The work by Lim, Kim, and Kim (2016) is similar to our work in that they study how to transfer from a speech corpus to a sound event corpus. Our work differs in two ways: first, we show results on deep convolutional networks up to 169 layers as opposed to Multilayer Perceptrons up to 6 layers, and second, we also propose a new multiscale representation based on dilated convolutions and show its positive interaction with transfer learning. We believe this work is the first to investigate cross-domain (speech commands vs. environmental sounds) transfer learning for modern network architectures.

Datasets

In our experiment, we utilize two datasets: UrbanSound8K, a dataset of 10 categories of environmental sounds (Salamon, Jacoby, and Bello 2014) and Google Speech Commands, a dataset of 30 categories of spoken terms (Warden 2017). Both datasets are collections where each audio clip represent a single class—types of common urban sounds for UrbanSound8K and single word speech utterances for the Speech Commands dataset. In this work, we perform several transfer learning experiments and in these experiments, UrbanSound8K serves as the source dataset and Google Speech Commands is the target dataset.

UrbanSound8K

The UrbanSound8K dataset, originally derived from the FreeSound[1] collection, consists of 8372 audio samples belonging to 10 categories – *air_conditioner*, *car_horn*, *children_playing*, *dog_bark*, *drilling*, *engine_idling*, *gun_shot*, *jackhammer*, *siren*, and *street_music*. Most audio samples are limited to 4 seconds long. The dataset comes partitioned into 10 folds for cross validation purposes. Audio samples are also labeled with their "salience"—a binary label denoting whether they were recorded in the foreground or background. While an interesting property, we did not explore how knowledge of salience could be used to improve model performance. This collection is quite challenging as many of the classes are highly confusable, even to a human ear, like *jackhammer* and *drilling* or *engine_idling* and *air_conditioner* due to the high timbre similarity, and the classes *children_playing* and *street_music* due to presence of complex harmonic tones. The UrbanSound8K dataset was created with a balanced distribution across the classes.

[1] freesound.org

Google Speech Commands

The Google Speech Commands dataset is an order of magnitude larger than the UrbanSound8K dataset and completely different in nature (environmental sounds vs. speech). The dataset is a crowd-sourced collection of 47,348 utterances of 20 short words—*yes*, *no*, *up*, *down*, *left*, *right*, *on*, *off*, *stop*, and *go*. The dataset was gathered by prompting people to speak single-word commands over the course of a five minute session, with most speakers saying each of them five times. The dataset also includes 17,373 samples from 10 non-command words. These words, like *bed*, *bird*, *cat*, *dog*, *happy*, etc., are unrelated to the core commands and are added to help distinguish unrecognized words. Unlike the core command words, the non-command words were said at most once by the speakers. Table 1 summarizes both datasets.

Feature Extraction

To prepare the audio data for neural network consumption, each audio file was processed with the following sequence of steps to do minimal feature extraction. First, the audio is re-sampled to 22kHz mono and partitioned into overlapping frames. The frames are 46 ms long and have 50% overlap (23 ms). This is in-line with the feature extraction protocol used by Salamon and Bello (2016). Then, the Mel spectrum is extracted using a Fourier transform and a Mel filter bank with 64 filters. The preprocessing pipeline is created using the Yaafe audio processing library (Mathieu et al. 2010) and results in each audio clip transformed into a sequence of frames with each frame being a 64-dimensional feature vector. In addition to the Yaafe preprocessing pipeline, we normalize the feature dimensions by subtracting their mean and dividing by their variance. In our experiments, we found this input normalization nonnegotiable.

Models

In this work, we apply modern convolutional neural networks to audio spectrograms. While seemingly straight forward, there are many methodological considerations that, although explored by the Computer Vision community, have not been as extensively reported for audio spectrograms. We investigate these considerations by experimentally varying two important properties of modern convolutional networks: network depth using skip connection techniques and multiscale input resolution using dilated convolutions.

Dilated Convolutions for Multiscale Inputs

Convolutional operations, intuitively, are windowed operations that scan over an input tensor. The free parameters of these operations are the size of the window (called the *kernel size*) and the step size of the scan (called the *stride*). The kernel size and stride parameterize the receptive field of the convolution: they control how much each convolution operation "sees" which allows for designing certain types of information flow.

More recently, a parameter referred to as *dilation* has been introduced as a way to increase the receptive field without increasing the number of parameters of the convolutional kernel (Yu and Koltun 2015). A dilation is, intuitively, a stride in the kernel—it is a spacing between the scalars in the kernel such that when it is scanned across an input tensor, the kernel subsamples a wider range of input values. This is visualized in Figure 1. More formally, consider a single position in the output tensor, $Y_{m,n}$. A convolution operation computes this value by summing over element-wise multiplications, as shown in Equation 1. In contrast, a dilated convolution, shown in Equation 2, sums element-wise multiplications that are d steps apart.

$$Y_{m,n} = \sum_{i=0}^{k} \sum_{i=0}^{k} W_{i,j} * X_{m+i,\, n+j} \quad (1)$$

$$Y_{m,n} = \sum_{i=0}^{k} \sum_{j=0}^{k} W_{i,j} * X_{m+i*d,\, n+j*d} \quad (2)$$

The core methodological contribution of this paper combines multiple convolutions with different dilation values into a single layer in order to fuse multiple scales of information. Specifically, we combine the outputs of four convolutional kernels with dilations of 1, 2, 3, and 4, a kernel size of 3 (both width and height are 3), and a stride of 1. By using a padding operation which reflects the kernel size and dilation of each convolution[2], the resulting output tensors are the same size and can be stacked along the channel dimension. To summarize, multiple convolutions with varying dilation and padding values are used to compute two-dimensional feature maps across a spectrogram, and subsequently stacked to form a stack of feature maps which can be used as input to another convolutional layer.

Increasingly Deeper Convolutional Networks

Several techniques have surfaced in recent years which enable dramatically deeper convolutional neural networks. In this study, we investigate the effectiveness of these techniques on classifying audio spectrograms. Specifically, we use two architectures, Residual Networks (ResNets) and DenseNets, which employ different techniques to achieve network depth.

ResNets Building on the traditional feed-forward architecture, ResNets (He et al. 2016) add a residual connection that allows the output of one layer to skip one or more layers before being summed with the output of another layer. Consequently, the computation at a single layer can potentially use the output of all previous layers and not just the immediately preceding layer. More formally, let F_l represent the computation of a layer at depth l and x_{l-1} represent the output of computation at layer $l-1$. In this notation, the traditional feed-forward network performs a sequence of operations such that $x_l = F_l(x_{l-1})$. With ResNets, a skip connection is added so that the computation of x_{l-1} is summed with the computation of $F_l(x_{l-1})$:

$$x_l = F_l(x_{l-1}) + x_{l-1} \quad (3)$$

[2]A 3x3 kernel is dilated with $d = 2$ and $padding = 2$ is used to ensure the output tensor is the same size as the input tensor

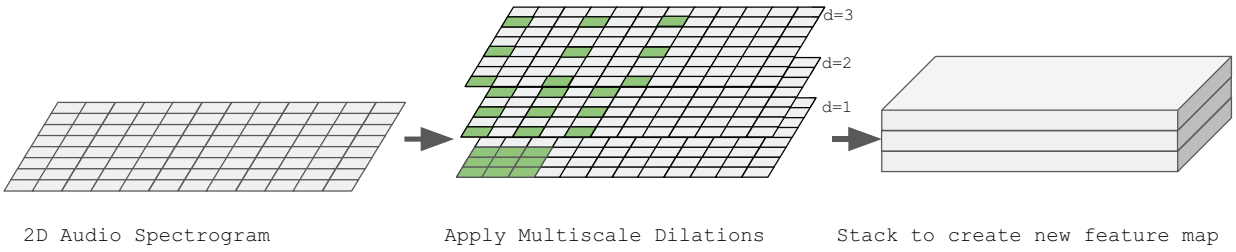

Figure 1: Starting with an audio spectrogram, we employ a set of dilations at different scales and with equivalent padding to produce new features maps which can be treated as stacked channels. Shown here are the first three dilations (d=1, d=2, d=3) with the fourth not shown to save space.

There are three ResNet models used in this work: ResNet-18, ResNet-34, and ResNet-50.

DenseNets In addition to ResNets, we use the DenseNets convolutional network architecture because of its state of the art performance and novel use of skip connections. DenseNets (Huang et al. 2016) were built upon a simple observation: convolutional networks greatly benefit from shorter connections between layers closer to the input and layers closer to the output. In more formal notation, the computation for x_l is dependent on the computations of all previous layers and all downstream layers have direct access to the feature maps of all earlier layers:

$$x_l \; = \; F_l(x_{l-1},\, x_{l-2},\; \ldots,\, x_0) \tag{4}$$

There are three DenseNet models that are evaluated in this work: DenseNet-121, DenseNet-161, and DenseNet-169.

Adapting Convolutional Network Models Traditional convolutional network architectures have been constructed and validated in Computer Vision settings. As a consequence, there are two modifications required to adapt image-specific convolutional network models to audio spectrograms. The first modification is the simpler change of adapting the number of channels in the first-layer to correspond to the single channel mono audio spectrogram. The second modification addresses each network's assumption that the input is a fixed input size. Since convolutional neural networks are designed to reduce a fixed input size to a fixed output size, violating the fixed input size requires solving the fixed output size problem. This was addressed by replacing the last max/average pooling layer with one that dynamically matches the length and width of the tensor that is output of the network.

Baseline Model

To provide a baseline for the modeling choices presented in this work, we implement the current state-of-the-art on the UrbanSound8K dataset, a convolutional network model called SB-CNN and proposed by (Salamon and Bello 2016).SB-CNN has three layers of convolutions, interspersed with max-pooling operations. The output of the third layer is flattened and two fully connected layers are then applied to result in a distribution over UrbanSound8K's label set. SB-CNN's design is very similar to traditional feed forward convolutional networks and is a useful comparison for the other models in this study.

Experiments and Results[3]

In this work, we conducted three experiments to measure the effects of convolutional neural network architectures, multiscale inputs, and transfer learning on classifying audio spectrograms. For the first experiment, we evaluated the choice of convolutional network architecture and the use of multiscale inputs on classifying environment sounds. In the second experiment, we selected and held constant the DenseNet-121 convolutional network architecture in order to measure the effects of multiscale inputs and the use of transfer learning on classifying speech commands. Finally, we repeat the second experiment but add a third factor by ablating the amount of data available for training in order to fully gauge the effectiveness of multiscale inputs and transfer learning.

Convolutional Networks on Environmental Sounds

The first experiment has two critical factors: the convolutional network architecture (SB-CNN, ResNet-18, ResNet-34, ResNet-50, DenseNet-121, DenseNet-161, and DenseNet-169) and multiscale inputs (with or without multiscale input). Each combination of architecture and multiscale input is instantiated and evaluated on the UrbanSound8K dataset.

Training Details For this experiment, we use the 10-fold cross-validation specified by the UrbanSound8K dataset (Salamon, Jacoby, and Bello 2014). Specifically, the model is trained on 8 of the 10 folds, validated on the 9th at the end of every epoch to determine when the training algorithm should terminate, and after training terminates, the model is evaluated on the remaining fold. Per standard cross-validation procedure, this is repeated 10 times with a different fold serving as the final evaluation fold in each repetition.

[3]This version of the paper reports better results on the Google Speech Commands dataset than an earlier version. The difference was primarily due to switching optimization algorithms from SGD to Adam.

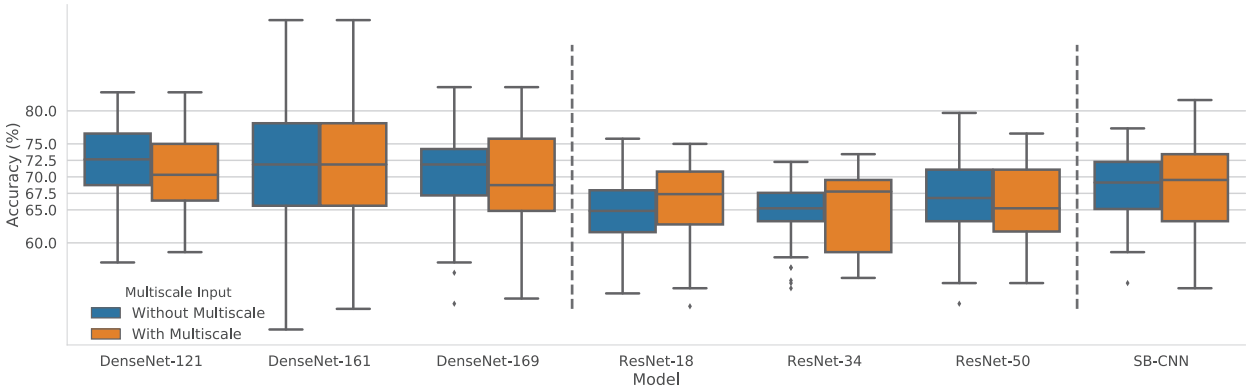

Figure 2: The accuracy on the UrbanSound8k dataset—aggregated over 10-fold cross validation—is shown as box plots for the convolutional network architectures and multiscale inputs. The body of each box plot denotes the 25th and 75th percentiles, the line in the body is the median, and the whiskers mark the most extreme observations.

	Without Multiscale	With Multiscale
SB-CNN	68.51±5.40	68.64±6.46
ResNet18	64.53±5.41	66.33±4.73
ResNet34	64.62±4.65	65.25±6.15
ResNet50	67.34±5.40	69.35±5.97
DenseNet121	72.61±5.14	70.82±5.48
DenseNet161	71.57±8.10	71.12±8.73
DenseNet169	70.33±6.96	69.56±7.45

Table 2: The micro-averaged accuracy and its standard deviation on the UrbanSound8K are shown for each of the 7 models and 2 multiscale conditions that were evaluated.

We further augment this setup by using early stopping mechanism that terminates training when performance on the validation fold has not improved for 10 epochs[4]. The model parameters from the best performing epoch—as measured on the 9th fold—are reloaded and used to evaluate final test set performance. The remaining critical hyper parameters are reported in Table 5.

Results The results are presented in Table 2 and Figure 2, but can be summarized with the following observations. First, the DenseNet architectures—which have been shown to be state-of-the-art for Computer Vision tasks—out perform ResNet and the SB-CNN baseline. Next, ResNet did not perform as well as expected and did not out perform SB-CNN on most comparisons. As a side note, despite our careful effort in closely following the description in Salamon and Bello (2016), we were unable to reproduce the SB-CNN accuracy of 73% as reported by the authors[5]. These results suggest that the kinds of skip connections in DenseNet provide a more robust model, although such a complex model might not be needed and SB-CNN could provide enough performance in certain circumstances.

[4]The choice of 10 was derived empirically by exploring 5, 10, and 15 as the number of epochs to wait.

[5]Our code, experiment notebooks, library versions, environment details, and hyperparameter setting details are available at: http://bit.ly/r7aaai18

	No Multiscale	
	Fresh Initialization	Pretrained
left vs. *right* Subset	96.70±1.41	97.09±1.16
All 30 Terms	91.48±1.67	91.63±1.95

Table 3: Classification performance for the pre-trained and non-pre-trained models without multiscale input on both sets of the speech commands dataset and the *left* vs. *right* subset.

	Multiscale	
	Fresh Initialization	Pretrained
left vs. *right* Subset	97.05±1.27	97.31±1.39
All 30 Terms	91.23±1.72	92.15±1.71

Table 4: Classification performance for the pre-trained and non-pre-trained models with multiscale input on both sets of the speech commands dataset and the *left* vs. *right* subset.

Transfer Learning for Speech Commands

In the second experiment, we evaluated how well transfer learning works from environmental sounds to speech commands both with and without multiscale inputs. To better assess the quality of learning, the speech command classification task was broken into two separate tasks of increasing difficulty: discriminating between the commands *left* vs. *right* and discriminating between all 30 short utterance categories. In total, there were six conditions in a factorial design: freshly initialized vs pre-trained network, multiscale input vs no multiscale input, and two versions of the dataset.

Training Details To isolate the effects of transfer learning and multiscale inputs, the DenseNet-121 convolutional network architecture was used for all conditions. In the transfer learning conditions, pre-trained network parameters which

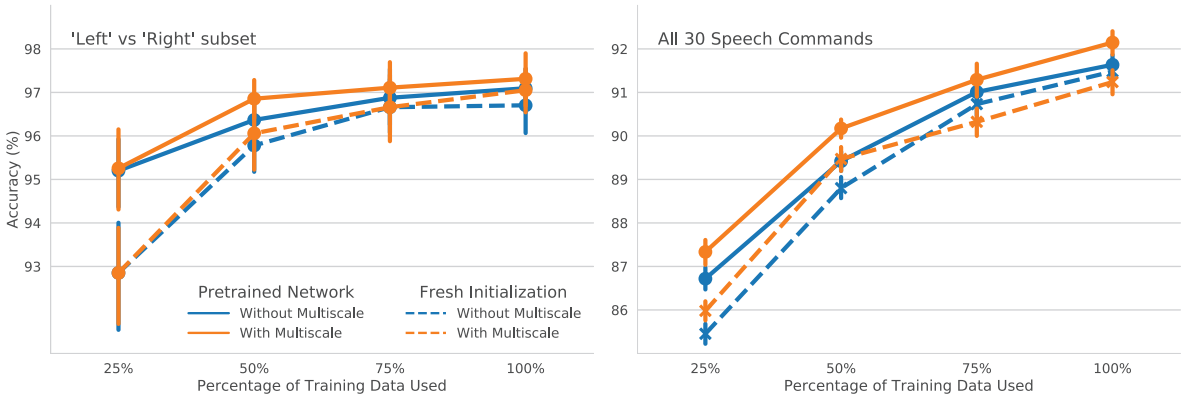

Figure 3: For transfer learning, the DenseNet-121 architecture is trained on only the *left* vs. *right* subset of the Google Speech Commands dataset using 25%, 50%, 75% or 100% of the data, with or without multiscale input, and with or without being pre-trained on UrbanSound8k.

were learned during the first experiment are loaded. Then, the final linear layer is replaced with one which matches the number of output classes needed for either the full speech command dataset or the *left* vs *right* subset. The remaining critical hyper parameters are reported in Table 5.

Results The results, shown in Tables 3 and 4, can be summarized with the following observations. First, the performance on the Google Speech Commands dataset is fairly high—around 91% for the non-pre-trained model with no multiscale inputs. This suggests that, despite the size, the dataset might be easy enough to get high performance. Next, the pre-trained networks have a significant increased performance ($p < 0.0001$) over networks that started with freshly initialized parameters. Additionally, there is an small interaction between the pre-training and multiscale such that the performance is highest under this condition. Finally, it's an interesting point to note that despite being an order of magnitude larger than the UrbanSound8K dataset, the Google Speech Commands dataset still benefited from learning transfer using the pre-trained representations from an unrelated classification task. This result is compelling because it suggests a strong interaction effect between the pre-training and multiscale dilated convolutions and warrants further investigation.

Transfer Learning and Target Dataset Ablation

In the final experiment of this study, we further evaluated the effectiveness of transfer learning and multiscale inputs by varying the amount of target training data available. Limiting the amount of target training data gives a sense of how well the pre-trained network generalizes. This is largely intended to better understand the interaction effects between multiscale inputs and pre-training.

Training Details In contrast with the second experiment where 100% of the target training dataset was used, the target dataset is ablated to either 25%, 50%, or 75% of its total. The ablations are selected using a fixed random number seed and repeated with 2 different seeds. Results are reported for the *left* vs. *right* subset of speech commands and on the whole dataset of 20 commands and 10 non-commands.

Results The results are shown in Figure 3 for the whole dataset and the *left* vs. *right* subset. There are several key take-away points. First, for the *left* vs. *right* subset, using only 75% of the training data, the pre-trained network obtained the nearly same performance as the freshly initialized network with 100% of the training data. The amount of training data drops further to 50% when the multiscale input representation is used in conjunction with pre-training. Next, there is a consistent synergistic effect between pre-training and multiscale inputs. The final take-away point is that the benefits of multiscale inputs are much lower in the freshly initialized networks. This is strong evidence of the interaction between pre-training and multiscale inputs: using pre-trained multiscale input through dilated convolutions prominently increases the transfer capabilities of the network.

Discussion

In this paper, we have analyzed modern convolutional neural network architectures for classifying audio spectrograms. By systematically enumerating the networks, the choice to use multiscale input, and the use of transfer learning, we have illuminated several key lessons.

Lessons The first lesson informs which convolutional network architecture should be used for classifying audio spectrograms. In Figure 4, the convolutional network architectures are plotted according to their training time, number of parameters, and accuracy. Overall, the evidence suggests that DenseNet architectures provide the best accuracy vs. model size vs. training time tradeoff for audio classification tasks.

The second lesson is that transfer learning from environmental sounds to speech commands has great potential. Not only were the pre-trained networks able to obtain higher accuracies with smaller subsets of the target data, but this pat-

	Optimizer	Learning Rate	Weight Decay	Dataset Split
Convolutional Networks on Environmental Sounds	Adam	0.001	1e-5	10-fold CV
Transfer Learning for Speech Commands	Adam	0.001/0.005	1e-4	60% / 20% / 20%
Transfer Learning and Target Dataset Ablation	Adam	0.001/0.005	1e-4	60% / 20% / 20%

Table 5: The critical hyper-parameters from our three experiments. Where two learning rates are indicated, the rates were used during transfer learning, and the larger rate was used for the newly initialized layers while the smaller rate was used for the pre-trained network parameters.

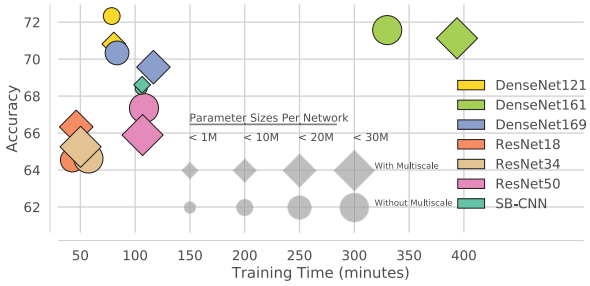

Figure 4: The classification accuracies for the convolutional network architectures—both with and without multiscale inputs using dilated kernel—are shown as a function of model size (number of parameters) and average training time.

tern was amplified for pre-trained networks which used the multiscale inputs. This suggests that the multiscale dilated convolutions could be learning important patterns and regularities in sound identification that transfer well to speech command classification.

Future Work The study presented in this paper serves as a starting point from which several intriguing approaches could be pursued. Our model performances, while near state of the art, are not at the level of human performance[6]. One promising approach is to evaluate how data augmentation interacts with our findings on pre-training and multiscale input with dilated kernels.

Another encouraging follow-up is to further investigate the impact of multiscale inputs. While this study observed that the multiscale input using dilated convolutions improved classification performance on freshly initialized networks and compounded the effectiveness of transfer learning, there are potential conflations that should be carefully studied and ruled out. In general, though, the promising results of multiscale inputs using dilated convolutions suggest that they can be combined with other techniques and warrants further study.

Conclusion

In this work, we have presented a study of convolutional network architectures applied to classifying the audio spectrograms of the UrbanSound8k and Google Speech Commands datasets. Our contributions are an exposition into the relationship between convolutional network architectures and audio spectrogram data, a novel multiscale input using dilated convolutions, and an examination of how well learning can transfer from an environment sounds dataset to a speech commands dataset.

We conclude by summarizing these contributions in five findings. First, we find that DenseNets provide the best accuracy/model size/training time tradeoff on audio spectrogram classification when compared to the baseline model, and several other skip connection models, including ResNet. Second, our novel use of dilated convolutions for multiscale inputs resulted in increased performance on audio classification tasks.

The next three findings are centered on the transfer learning experiments. To begin, our third finding is that convolutional networks pre-trained on environmental sound classification out-performed freshly initialized convolutional networks on the task of classifying speech commands. As a consequence, our fourth finding is that this gap in performance means we can obtain the same classification accuracy with less training data. Finally, our fifth finding is that the previous two findings are even stronger when multiscale inputs through dilated convolutions are employed. In other words, pre-training on environmental sounds with a convolutional network that utilizes multiscale inputs through dilated convolutions can substantially increase classification accuracy with a fraction of the data.

Through this study, we have evaluated a series of convolutional network architectures and different modeling choices on audio spectrograms. Further, we have demonstrated a relationship between the kinds of representations needed for recognizing environmental sounds and for recognizing speech commands. Moving forward, there are many promising directions which can further unify audio event identification for both human speech and ambient environmental sounds.

[6]Reported as 82% on 4 second environmental sound clips (Chu, Narayanan, and Kuo 2009).

References

Choi, K.; Fazekas, G.; Sandler, M. B.; and Cho, K. 2017. Transfer learning for music classification and regression tasks. In *International Society for Music Information Retrieval Conference*.

Choi, K.; Fazekas, G.; and Sandler, M. 2016. Automatic tagging using deep convolutional neural networks. *arXiv preprint arXiv:1606.00298*.

Chu, S.; Narayanan, S.; Kuo, C.-C. J.; and Mataric, M. J. 2006. Where am I? scene recognition for mobile robots using audio features. In *Multimedia and Expo, 2006 IEEE International Conference on*, 885–888. IEEE.

Chu, S.; Narayanan, S.; and Kuo, C.-C. J. 2009. Environmental sound recognition with time-frequency audio features. *Transactions in Audio, Speech, and Language Processing* 17(6).

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, 248–255. IEEE.

Dieleman, S., and Schrauwen, B. 2014. End-to-end learning for music audio. In *ICASSP, 2014 IEEE International Conference on*, 6964–6968. IEEE.

Hamel, P.; Davies, M. E.; Yoshii, K.; and Goto, M. 2013. Transfer learning in mir: Sharing learned latent representations for music audio classification and similarity. In *ISMIR*, 9–14.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

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

Huang, G.; Liu, Z.; Weinberger, K. Q.; and van der Maaten, L. 2016. Densely connected convolutional networks. *arXiv preprint arXiv:1608.06993*.

Inkyu Choi, Kisoo Kwon, S. H. B., and Kim, N. S. 2016. Dnn-based sound event detection with exemplar-based approach for noise reduction. In *Proceedings of Detection and Classification of Acoustic Scenes and Events*.

Lee, J., and Nam, J. 2017. Multi-level and multi-scale feature aggregation using pre-trained convolutional neural networks for music auto-tagging. *arXiv preprint arXiv:1703.01793*.

Lim, H.; Kim, M. J.; and Kim, H. 2016. Cross-acoustic transfer learning for sound event classification. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, 2504–2508. IEEE.

Maijala, P.; Shuyang, Z.; Heittola, T.; and Virtanen, T. 2018. Environmental noise monitoring using source classification in sensors. *Applied Acoustics* 129:258 – 267.

Mathieu, B.; Essid, S.; Fillon, T.; and Prado, J. 2010. Yaafe, an easy to use and efficient audio feature extraction software. In *Proc. of the 11th Int. Conf. on Music Information Retrieval (ISMIR*.

Miller, D. R.; Kleber, M.; Kao, C.-L.; Kimball, O.; Colthurst, T.; Lowe, S. A.; Schwartz, R. M.; and Gish, H. 2007. Rapid and accurate spoken term detection. In *Eighth Annual Conference of the International Speech Communication Association*.

Ntalampiras, S.; Potamitis, I.; and Fakotakis, N. 2009. An adaptive framework for acoustic monitoring of potential hazards. *EURASIP Journal on Audio, Speech, and Music Processing* 2009.

Oord, A. V. D.; Dieleman, S.; Zen, H.; Simonyan, K.; Vinyals, O.; Graves, A.; Kalchbrenner, N.; Senior, A. W.; and Kavukcuoglu, K. 2016. Wavenet: A generative model for raw audio. *CoRR* abs/1609.03499.

Parlak, S., and Saraclar, M. 2008. Spoken term detection for turkish broadcast news. In *ICASSP 2008. IEEE International Conference on*, 5244–5247. IEEE.

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

Piczak, K. J. 2015b. Esc: Dataset for environmental sound classification. In *Proceedings of the 23rd ACM international conference on Multimedia*, 1015–1018. ACM.

Pons, J.; Slizovskaia, O.; Gong, R.; Gómez, E.; and Serra, X. 2017. Timbre Analysis of Music Audio Signals with Convolutional Neural Networks. *ArXiv e-prints*.

Sainath, T. N., and Parada, C. 2015. Convolutional neural networks for small-footprint keyword spotting. In *Sixteenth Annual Conference of the International Speech Communication Association*.

Salamon, J., and Bello, J. P. 2016. Deep convolutional neural networks and data augmentation for environmental sound classification. *CoRR* abs/1608.04363.

Salamon, J.; Jacoby, C.; and Bello, J. P. 2014. A dataset and taxonomy for urban sound research. In *22st ACM International Conference on Multimedia (ACM-MM'14)*. ACM.

Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; and Wojna, Z. 2015. Rethinking the inception architecture for computer vision. *CoRR* abs/1512.00567.

Temko, A.; Malkin, R.; Zieger, C.; Macho, D.; Nadeu, C.; and Omologo, M. 2006. Clear evaluation of acoustic event detection and classification systems. In *International Evaluation Workshop on Classification of Events, Activities and Relationships*, 311–322. Springer.

Wang, Y., and Metze, F. 2017. A transfer learning based feature extractor for polyphonic sound event detection using connectionist temporal classification. In *InterSpeech*.

Warden, P. 2017. *Speech Commands: A public dataset for single-word speech recognition*.

Xu, Y.; Du, J.; Dai, L.-R.; and Lee, C.-H. 2014. Cross-language transfer learning for deep neural network based speech enhancement. In *ISCSLP, 2014 9th International Symposium on*, 336–340. IEEE.

Yu, F., and Koltun, V. 2015. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*.

Zeiler, M. D., and Fergus, R. 2014. Visualizing and understanding convolutional networks. In *European conference on computer vision*, 818–833. Springer.