From Image Classification to Audio Classification

Yogesh Kumar*
140050004
IIT Bombay
yogesheth@cse.iitb.ac.in

Mohit Vyas*
140050015
IIT Bombay
mohitvyas@cse.iitb.ac.in

Saurabh Garg*
140070003
IIT Bombay
saurabhgarg@cse.iitb.ac.in

Abstract

Transfer Learning based paradigms which tends to use learned representations from standard image classification problem forms the state-of-the-art for Audio Classification tasks. In this project, we adopted a Convolutional Neural Network based approach for Audio Classification. Using pre-trained CNNs enables us to use Transfer learning. We implemented and explored models which uses embeddings generated from standard CNN architectures for image processing, namely VGG and ResNet to perform Acoustic Event Detection(AED)/ classification task. Further, we use a simple multiscale input representation using dilated convolutions which aggregates larger contexts and increase classification performance. The models trained using multiscale inputs and transfer learning generalize across datasets. Our analysis is supported by experimental results on Audio Set and UrbanSound 8k for audio classification. We further use trained VGG based network to generate classification summary on a football match commentary and a Ted-Talk.

1 Introduction

Although speech is certainly the most informative acoustic event, other kinds of sounds may also carry useful information. Detection and classification of such sounds is essential for understanding human and social activity in an environment. Effectively detecting such sounds could make the current ASR systems more robust by identifying the non speech parts of a sound. Acoustic event detection aims to characterize the acoustic environment of an audio stream by selecting a semantic label for it. It can be considered as a machine-learning task within the widespread multi-label classification paradigm, in which a set of class labels is provided and the system must select a subset of labels for any given input.

Over the past few years, the performance of Acoustic event detection has been significantly increased due to the advances in deep Convolutional Neural Networks and Sequence to Sequence models. However, to the best of our knowledge, there has been little work on determining where in the audio file did the event occurred. The current methods also fail to generalize on other datasets.

In this project, we explore a novel method to tag the audio events to timestamps. The task is analogous to the object detection task for video files. We used the current state-of-the-art deep convolutional networks to extract features from spectrograms, which are then classified using a DNN/RNN on a multi-label classification task.

*All authors contributed equally

2 Motivation

Our main motivation for the project was to do Audio Classification by transferring the learned representation from various successful architectures in image classification. There have been huge success in image classification and related problems like summary generation, object detection, etc., but not many things are explored for Audio Classification and tasks like keyword spotting or audio based searches. Not only this, knowledge of whether an audio segment contains speech or some other background noise can help to save tremendous human labor to segment audio files. Further, the information of whether an audio segment contains a specific class (say cheering, applause or siren) provides crucial insights on the information contained in the segment without Natural Language Understanding.

3 Task Definition

Given an audio clip the aim is to classify the audio segments into set of predefined classes. This is a multi-class multi-label classification problem in general as an a segments can contain overlapping fragments of sound from many classes. It is essentially an acoustic event detection/classification task.

Mathematically, the problem is given $\mathcal{X} = \{x_1, x_2, ..., x_t\}$ the task to generate labels $\mathcal{Y} = \{y_1, y_2, ..., y_t\}$ such that each $y_i \subset \mathcal{C}$ where $\mathcal{C}$ is the set of all classes.

We extract log-mel spectrogram of the audio input. Thus, the problem we solve is to classify log-mel spectrogram of the given audio segment into set of predefined classes.

4 Dataset(s) Used

4.1 Audioset

The AudioSet consists of an expanding ontology of 632 audio event classes and a collection of 2,084,320 human-labeled 10-second sound clips drawn from YouTube videos. The ontology is specified as a hierarchical graph of event categories, covering a wide range of human and animal sounds, musical instruments and genres, and common everyday environmental sounds.

We crawled relevant classes from the dataset. Namely, we used Clapping, Cheering, Applause, Chatter, Children Playing, Shout, Whispering, Laughter, Crying and Speech as our classes, as we wanted to use this in downstream applications concerned with audio summarization and segmentation of speech vs non-speech data. In this set of classes we made sure that there are sufficient confusing classes like speech is confusable with chatter.

On AudioSet, the problem is to perform the defined multi class classification as one audio segments can corresponds to more than one class from the set of predefined classes.

4.2 UrbanSound 8K

The UrbanSound 8K dataset 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. 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.

On UrbanSound 8K, the problem is perform single class classification as one audio segments corresponds to only one class from the set of predefined classes.

5 Methodology

5.1 Models Explored

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, VGG and Residual Networks (ResNets), which employ different techniques to achieve network depth.

5.1.1 VGG

We use a slightly modified VGG architecture[6]. This network is characterized by its simplicity, using only 3×3 convolutional layers stacked on top of each other in increasing depth. Reducing volume size is handled by max pooling. Two fully-connected layers, each with 4,096 nodes.

5.1.2 ResNet-50

Building on the traditional feed-forward architecture, ResNets[2] uses the concepts of skip-connections to build deep CNNs. Further, we added dilated convolutions in the original ResNet architecture. We discuss the concepts of skip-connections and dilation before describing the network architecture.

Skip Connections

Very deep convolutional architectures suffers from the problem of "degradation". More specifically, as a network starts converging, with the network depth increasing, accuracy gets saturated and then degrades rapidly. Unexpectedly, such degradation is not caused by overfitting, and adding more layers to a suitably deep model leads to higher training error. This motivated to 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. So, the output of a layer can theoretically depend on the output of all the previous layers and not just the preceding layer. Intuitively, this will make network partially shallow and may help mitigate this issue originating from network being very deep. 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 $l-1$. Then, 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 $F_l(x_{l-1})$:

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

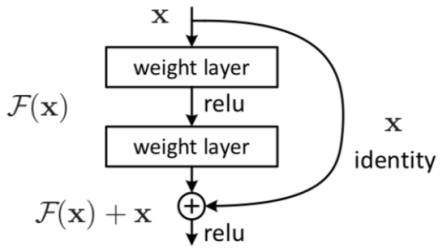


Figure 1: a visualization of skip connections

Model Architecture

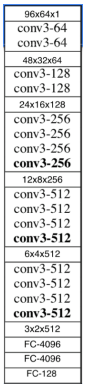
Model consist of 2-layer building blocks with kernel size 3x3 and number of filters 64, 128, 256 and 512. There is a skip connection between input and output of each building block.

5.2 Dilated Convolutions

We want our convolutional network to integrate information from different temporal scales and balance two properties:

layer name	output size	18-layer
conv1	112×112	7x7, 64, 2
conv2_x	56×56	3x3 maxpool, 2 $\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$

(a) ResNet Model description



(b) VGG Model description

Figure 2: Model Architectures

- local, frame level accuracy, detecting a sudden spike
- integrating knowledge of the wider, global context

One of the possible ways to integrate these properties is spatial pooling, but it leads to a huge increase of the number of parameters. Motivation behind Dilated-Kernels is to increase the receptive field without increasing the number of parameters of the convolutional kernel. 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 (with no increase in number of parameters). 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:

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

A dilated convolution, however, has a strided kernel such that the positions in the input tensor are spaced further apart:

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

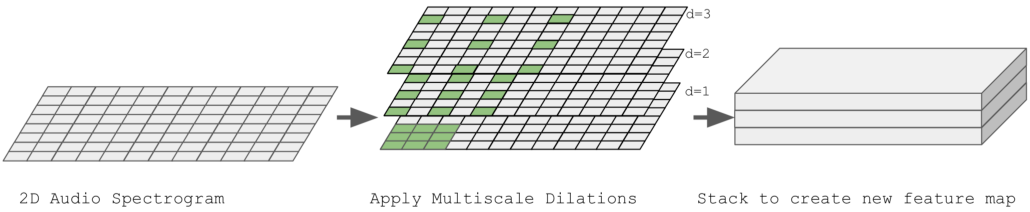


Figure 3: Picture visualizes how dilation increases the receptive field of kernel for d= 1, 2 and 3

6 Implementation Details

6.1 Feature Extraction

Input feature map to ResNet architecture is generated using the following pipeline:

- Audio files are decomposed into 46 ms windows and 23 ms hop length and a short-time Fourier transform is applied

- The resulting spectrogram are then integrated into 64 mel-spaced frequency bins, and the magnitude of each bin is log transformed
- This gives log-mel spectrogram patches of 435×64 bins for a 10 sec clip
- Outputs of four convolutional kernels with dilations of 1, 2, 3, and 4, a kernel size of 3x3, and a stride of 1 are combined to obtained dilated-features which are then fed into about ResNet architecture with 18 layers

For VGG architecture the features were generate in similar way with slight modification in mainly step 1. There each audio file is divided into non-overlapping 0.96s clips and The 960 ms frames are decomposed with a short-time Fourier transform applying 25 ms windows every 10 ms.

We implemented the ResNet Architecture in Tensorflow from scratch and modified code for pre-trained VGG net in Tensorflow to adept to our problem. We used librosa to extract features from the audio segments. For automatic speech recognition task we modified kaldi scripts to get word error rates on the default Test data by training the HMM-GMM model on 20% of the available training data.

7 Experimental Setup

We did the following experimentations and observations on these experiments are summarized in the next section.

- We explored and implemented transfer learning for audio classification using standard pre-trained architectures for image classification.
  - VGG (with DNN and RNN final layer)
  - ResNet (with and without dilations)
- To check the effect of removal of annotations (like Breath, Noise etc.) from transcriptions we ran ASR on tedlium dataset.
- To evaluate the efficacy we ran classification on an example of 7 min 28 sec football match commentary and on Ted Talk of 14 min 3 sec.

8 Experiments and Results

We trained the models discussed on 2 datasets, Audio-Set and UrbanSound8K, and respective details of experiments are discussed below. Dropout regularization and Adam optimizer is used in both the cases. Hyperparameters including dropout, batch size and learning rate is tuned using grid search whenever possible (otherwise single hyperparameter is tuned keeping others constant to reduce time taken).

8.1 AudioSet

This dataset comprised of approx. 4000 clips of 10 seconds each with frequency and classes as follows:

Significant Classes	Cheering	Chatter	Shout	Whisper	Laughter	Speech	Applause
Frequency	407	951	658	558	962	2953	88

Insignificant Classes	Children Playing	Crying	Clap
Frequency	207	82	758

Table 1: Summary of crawled Audioset data

Precision and recall are used as evaluation metrics and results on test-set are as follows.

AudioSet	**Micro Precision**	**Micro Recall**	**Micro F1-score**
With Dilation	0.66	0.64	0.654
Without Dilation	0.65	0.65	0.65

Table 2: Results on Audio set using ResNets

AudioSet	**Micro Precision**	**Micro Recall**	**Micro F1-score**
Pre-trained Model	0.8984	0.7998	0.8462

Table 3: Results on Audio set using VGGnets

8.2 UrbanSound8K

This dataset comprised of 8000 clips of approx. 4 seconds each. Classes include air conditioner, car horn, children playing, dog bark, drilling, engine idling, gunshot, jackhammer, siren, and street music. All classes have equal frequencies in the dataset.

Accuracies on test set are as follows.

Urban-Sound	**Accuracy**
With Dilation	70.8
Without Dilation	69.7

Table 4: Accuracy on UrbanSound 8K with ResNets

Urban-Sound	**Accuracy**
Pre-trained Model	59.51

Table 5: Accuracy on UrbanSound 8K with VGGnets

8.3 ASR on Tedlium

We compared WER of annotated and unannotated Ted talk. The results we obtained are as follows.

Word Error Rate	**Dev Set**		**Test Set**	
	Tri	Mono	Tri	Mono
With Annotations	23.6	31.6	23.0	32.9
Without Annotations	23.5	31.5	23.1	32.9

Table 5: Word Error rates on tedlium

We decided not to segment the tedlium data as unannotated accuracies were comparable (actually improvement of WER 0.1 on the dev set%).

8.4 Classification Results

We took real audio files namely a football match commentary and a ted talk and ran the VGG-Net trained on Audio Set to get classes all 1s segments.

Following tables summarize the output of classification.

Relevant Classes	**Cheering**	**Applause**	**Clapping**	**Speech**	**Shout**
Duration	152s	86s	463s	99s	55s

Irrelevant Classes	**Chatter**	**Laughter**	**Children playing**	**Whisp ering**	**Crying**
Duration	0s	205s	246s	94s	11s

Table 6: Summary of classes predicted on a Football Match Commentary [link]

Relevant Classes	**Cheering**	**Applause**	**Chatter**	**Speech**	**Clapping**
Duration	50s	67s	662s	116s	60s

Irrelevant Classes	**Laughter**	**Children playing**	**Shout**	**Crying**	**Whispering**
Duration	0s	1s	200s	31s	162s

Table 7: Summary of classes predicted on a Ted Talk [link]

There was lot of confusion observed between chatter and Speech classes in the ted-talk. Also clapping is highly confused with applause. Due to lack of training data available, at few places irrelevant classes were predicted even without any trace of those classes in the labels.

9 Conclusion

In the project we explored transfer learning to transfer learned representation from one problem to the other. Mainly,

- We explored how years of research done image-classification can be used to obtain reliable models for audio classification.
- Specifically we explored how pre-trained VGG model can be used for audio-classification and trained ResNet model from scratch with multi scaled inputs.
- Utilized successful image-classification architectures for the problem at hand and showed efficacy of the trained model on two datasets and on example classification task.

10 Future Work

As discussed in the motivation section, there are tremendous applications of audio classification problem. We would like to improve the system by training on bigger relevant dataset. Further, we would like to segments movie audios into speech and non-speech class and then evaluate of WER using state-of-the-art ASR systems. Comparable results, if obtained, can save huge amount of the human labor involved in annotating and segmenting the audio files.

References

[1] 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.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

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

[4] Brian McMahan and Delip Rao. Listening to the world improves speech command recognition. *arXiv preprint arXiv:1710.08377*, 2017.

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

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