Project Report

Md Shamim Hussain
Department of Computer Science
Rensselaer Polytechnic Institute
110 8th St.
Troy, NY, USA
hussam4@rpi.edu

Nafis Neehal
Department of Computer Science
Rensselaer Polytechnic Institute
110 8th St.
Troy, NY, USA
neehan@cs.rpi.edu

Abstract

Deep neural networks are notorious for their high computational cost and memory requirements. Not only that, often they need specialized hardware for efficient execution. However, for simple tasks such as a coarse-grained classification of environmental sounds into superclasses such as speech, music, and noise, one may not be able to spare a lot of computational resources. Such situations may arise when real-time audio classification/segmentation needs to be performed - e.g., in hearing aids, or at the frontend of real-time audio compression systems. So, we aim to devise network architectures that are fast and memory-efficient for this problem. This architecture should perform well on the CPU with limited memory requirements.

1 Introduction

Due to rapid increase in the amount of available audio data all over the world, time now demands an efficient method to automatically segment or/and classify audio stream based on its content. Audio classification typically refers to the assignment of the content of a given audio excerpt to a particular class. And audio segmentation refers to assigning different temporal regions of an audio content to different classes.

Music, speech and noise classification and segmentation is an important task because these three types of signals are inherently different in nature [9] and require different types of processing and/or coding schemes. For example, a good compression scheme for speech may not be good for compressing music. Also, the signal processing steps used for these two types of media are likely to be very different. So, correctly identifying the type of media before further processing should be considered a very important task.

On the advent of widespread use of the internet, it is now possible to build large media databases from user-contributed data. However, labels for the collected data are rarely available. Manual media labeling can be both expensive and time-consuming. So, a reliable method is required to automate the indexing of large media databases. It is intuitive to first classify/ segment the media into broad categories such as speech, music and noise which may be followed by further fine-grained classification/segmentation procedure (for example, into different musical genres).

The above discussion suggests that it is desirable to bring the classification/segmentation of media into speech, music and noise under a single framework. Engineering appropriate features for this problem can be both difficult and cumbersome. For example, a capella music has features similar to speech, whereas rock and roll music may have features similar to noise. So, we tried to solve this problem from general purpose features (such as MFCC and spectrogram) rather than artificially formulated task-specific features.

Preprint. Under review.

In recent years deep learning [10] algorithms have achieved unprecedented success in numerous classification problems, without any need for careful feature selection. Especially, deep convolutional neural networks have set new frontiers in many fields such as image and audio classification and segmentation. However, deep neural networks are generally known to be more computationally expensive and slower than other more conventional models. These models often need large-scale parallelization in GPUs for faster implementation. Since classification/segmentation is often set as a pre-processing step in an audio processing pipeline, it needs to be fast enough to prevent the delay in further processing. Also, the computational resources are limited in many scenarios such as in mobile devices and embedded systems where it is unreasonable to waste too much resource in the preprocessing step. This is where we put forth our contribution.

In this work, we present SwishNet, a carefully designed novel 1D convolutional deep neural network architecture which can achieve a high level of accuracy while also being fast, lightweight and memory efficient, even without large-scale parallelization in a GPU. Also, the deep convolutional nature of the architecture allows it to pick up contextual information from previous frames effectively. It can be trained on a sufficiently diverse dataset such as the MUSAN corpus without the need for any specific feature engineering. And in the presence of new data (e.g. new musical genres), it can simply be fine-tuned without the need for retraining from scratch.

2 Related Works

In general, audio content analysis in video parsing can be considered in two directions [5][6]. One is to discriminate audio streams into different classes such as speech, music, noise etc., the other is to classify audio streams into segments of different speakers. In this paper, our research work in the first direction will be presented.

There have been many studies on audio content analysis, using different features and different methods [4] -

Pfeiffer et al [1], presented a theoretic framework and application of automatic audio content analysis using some perceptual features. Saunders [2], presented a speech/music classifier based on simple features such as zero crossing rate and short time energy for radio broadcast. When a window size of 2.4s was used, the reported accuracy rate would be 98%.

Scheirer et al [3] introduced many more features into audio classification and performed experiments with different classification models including GMM (Gaussian Mixture Model), BP-ANN (Back Propagation Artificial Neural Network) and KNN (K-Nearest Neighbor). When using window of the same size (2.4s), the reported error rate would be 1.4%. However, it is found that such simple features-based methods cannot work well when smaller window is used or more audio classes such as environment sounds are taken into consideration.

Many other works have been done to enhance audio classification algorithms. In [7], audio recordings are classified into speech, silence, laughter and non-speech sounds, in order to segment discussion recordings in meetings. The accuracy of the segmentation resulted using his method varies considerably for different types of recording. In the work by Zhang and Kuo [8], pitch tracking methods are introduced to discriminate audio recordings into more classes, such as songs, speeches over music, with a heuristic-based model. Accuracy of above 90% is reported. Srinivasan et al [5], try to detect and classify audio that consists of mixed classes, such as combinations of speech and music together with background sound. The accuracy of classification is over 80%.

3 Methodology

3.1 Data Collection

The Audioset [9] ontology was used to identify fine-grained sound event classes (source). We combined the fine-grained event classes into 3 major superclasses - speech, music and noise. We define each superclass uniquely by saying that a particular audio segment must not contain events from other superclasses, so that there is no ambiguity in classification.

We downloaded the annotations of the YouTube video segments (source). Each segment is assigned multiple sound event tags. We defined plausible superclasses, based on these tags. If a clip contains

Table 1: Network Sizes and Prediction Times per Sample

Network	No. of Parameters	Prediction Time (ms) per Sample for Different Sample Lengths						Weight file size
		0.5s		1.0s		2.0s		
		CPU	GPU	CPU	GPU	CPU	GPU	
GMM	92,416	1.8	-	3.6	-	7.2	-	370KB
SNN	179,203	0.71	0.72	0.92	0.72	1.52	0.82	717KB
SwishNet-slim	5,483	0.77	1.42	0.88	1.43	1.12	1.45	22KB
SwishNet-wide	18,267	0.9	1.45	1.1	1.45	1.65	1.45	66KB
MobileNet	217,235	4.1	3.51	5.27	3.52	8.57	3.55	870KB

only the subclasses of a given superclass (speech/music/noise) it is said to belong to that plausible superclass. If the superclass is ambiguous, we ignore that clip. But due to mistakes in annotations, missing annotations and human errors, the actual superclass may still be ambiguous or even wrong. So, the labels assigned at this point are noisy.

Next, we downloaded YouTube clips in uncompressed wav format which was later compressed by the free lossless audio codec (FLAC). Clips were downloaded from the evaluation set, the balanced training set and the unbalanced training set. We downloaded about 81,000, 10s clips from YouTube.

To generate ground truth (gold standard) annotations for the clips we manually checked each clip and verified their plausible superclass. If we found that a clip's superclass was ambiguous, it was tagged as "bad", otherwise it was tagged as "good".

We performed these manual annotations until we reached 4,000 clips for each superclass (speech, music, and noise). Thus, the gold standard ("good") dataset contains 12,000 annotated unambiguous ground truth labeled clips. Also, the ambiguous ("bad") labels are also kept for possible future use in automated label clean up. Our annotated dataset, along with unchecked clips is available publicly (source).

3.2 Feature Extraction

3.3 Network Architecture

4 Results and Discussions

5 Conclusion and Future Work

Table 2: Overall and Speech/Non-Speech (SNS) Classification Accuracy for Clips of Different Lengths

Clip Length		**0.5s**		**1.0s**		**2.0s**	
Network		**Overall**	**SNS**	**Overall**	**SNS**	**Overall**	**SNS**
GMM		96.53%	98.58%	97.33%	99.05%	97.79%	99.33%
SNN		97.07%	98.87%	97.41%	99.13%	97.71%	99.36%
Swish-Net-slim	**Undistilled**	97.64%	99.24%	98.20%	99.60%	98.41%	99.76%
	Distilled	97.52%	99.19%	98.22%	99.51%	98.57%	99.70%
Swish-Net-wide	**Undistilled**	97.97%	99.37%	98.32%	99.67%	98.65%	99.75%
	Distilled	98.05%	99.45%	98.54%	99.71%	98.92%	99.84%
Mobile-Net	**Random**	98.13%	99.43%	98.53%	99.71%	98.95%	99.88%
	Pretrained	98.94%	99.73%	99.24%	99.89%	99.38%	99.96%

Table 3: Normalized Confusion Matrices for Clips of Different Lengths (Rows: True Labels, Columns: Predicted labels, Ordering: Noise, Music, and Speech)

Network	**0.5s**	**1.0s**	**2.0s**
GMM	$\begin{bmatrix} .79 & .19 & .02 \\ .02 & .96 & .01 \\ .01 & .01 & .98 \end{bmatrix}$	$\begin{bmatrix} .80 & .18 & .01 \\ .02 & .97 & .01 \\ .01 & .00 & .99 \end{bmatrix}$	$\begin{bmatrix} .79 & .18 & .02 \\ .01 & .98 & .01 \\ .00 & .00 & .99 \end{bmatrix}$
SNN	$\begin{bmatrix} .67 & .28 & .05 \\ .01 & .98 & .02 \\ .00 & .00 & 1.0 \end{bmatrix}$	$\begin{bmatrix} .67 & .28 & .05 \\ .00 & .98 & .01 \\ .00 & .00 & 1.0 \end{bmatrix}$	$\begin{bmatrix} .68 & .28 & .04 \\ .00 & .99 & .01 \\ .00 & .00 & 1.0 \end{bmatrix}$
SwishNet-slim (Undistilled)	$\begin{bmatrix} .78 & .19 & .03 \\ .01 & .98 & .01 \\ .00 & .00 & .99 \end{bmatrix}$	$\begin{bmatrix} .83 & .14 & .03 \\ .01 & .99 & .00 \\ .00 & .00 & 1.0 \end{bmatrix}$	$\begin{bmatrix} .84 & .15 & .01 \\ .01 & .99 & .00 \\ .00 & .00 & .99 \end{bmatrix}$
SwishNet-wide (Distilled)	$\begin{bmatrix} .81 & .17 & .02 \\ .01 & .99 & .00 \\ .00 & .00 & 1.0 \end{bmatrix}$	$\begin{bmatrix} .86 & .13 & .01 \\ .01 & .99 & .00 \\ .00 & .00 & 1.0 \end{bmatrix}$	$\begin{bmatrix} .88 & .11 & .01 \\ .01 & .99 & .00 \\ .00 & .00 & 1.0 \end{bmatrix}$
MobileNet	$\begin{bmatrix} .90 & .09 & .01 \\ .01 & .99 & .01 \\ .00 & .00 & 1.0 \end{bmatrix}$	$\begin{bmatrix} .91 & .09 & .01 \\ .00 & .99 & .00 \\ .00 & .00 & 1.0 \end{bmatrix}$	$\begin{bmatrix} .94 & .06 & .00 \\ .01 & .99 & .00 \\ .00 & .00 & 1.0 \end{bmatrix}$