AUTOMATIC SPEECH RECOGNITION
AUTUMN 2017

Course Project Report

Chitwan Saharia	150050011
Harshith Goka	150050069
Vishwajeet Singh	150050046

December 5, 2017

Contents

ABSTRACT

Convolutional Networks have been proven to be useful in categorizing images due to advances in computing power and availability of large scale data-set. The same network have been used to tackle the problem of speaker identification which is currently dominated by the I-vector approach in academia as well as commercial systems.

1 Introduction

Speaker Recognition is a well studied problem and can be broadly classified in two sub-problems. The first is speaker verification which aims to verify a given utterance and speaker-id using the already existing database and speaker model. The other is the topic of our interest and work which is speaker identification, which is aimed at figuring out the speaker from a given utterance. The specific task we are trying to look at is that of identifying the speaker from a closed set in a text-independent fashion. The data consists of utterances from various speakers in uncontrolled setting. The model has to be highly noise and variation resistant to be able to accurately predict the speaker identity. The main objective that we are trying to maximize is the top-1(%) accuracy. Present state of the art techniques in the this task are i-vectors, and neural networks have not been able to do well due to insufficiency of data. But with the advent of big speech datasets like VoxCeleb, we can now train neural networks to achieve better performance. Our goal is to leverage the amount of training data in VoxCeleb dataset and train CNN to classify the speakers directly. Our model attains 12% more accuracy compared to the state of the art i-vectors.

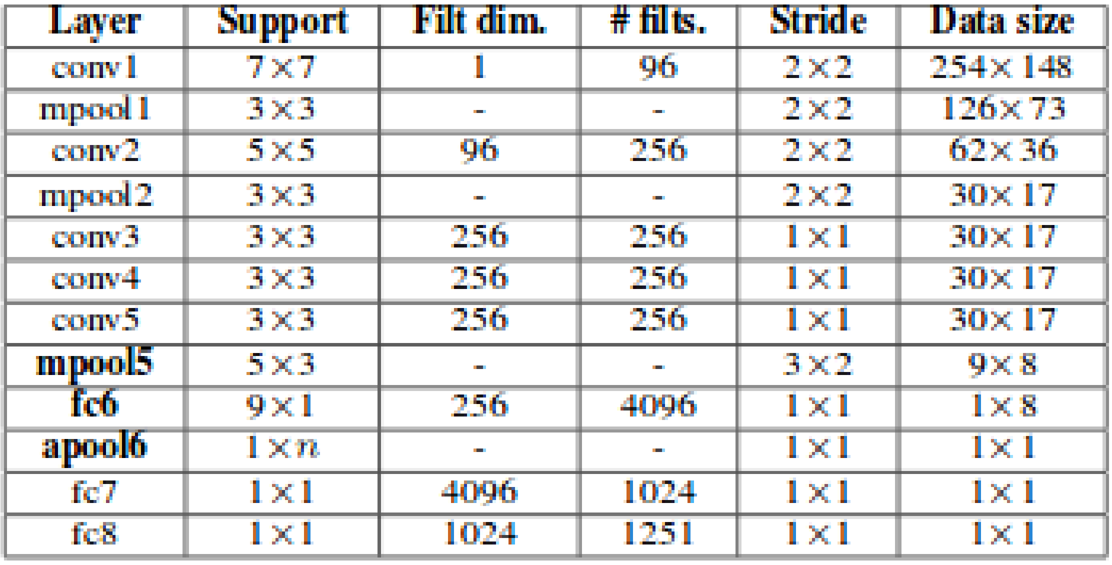
2 Methodology

2.1 Prior Work

I-vector is the current state of the art and is highly used in commercial systems. Recently, the time delayed neural network[1] have been used to extract I-vector and have outperformed the traditional extraction technique. The pre-cursor to I-vector was the joint estimation method which tries to model the speaker characteristics and the channel characteristics separately. I-vector tries to combine both of them to produce a significantly low dimension vector of the magnitude of hundreds only. Both the methods rely on a universal background model which is trained to extract the eigenvoices. After the I-vector extraction standard classification algorithm are used like SVM's, Neural Netowrks etc.

2.2 Input Features

The convolutional networks inherently works with fixed 2-dimensional inputs traditionally images with 1 or more channels. The MFCC's of the utterance provides a highly feature rich representation of the sample. The MFCC's plotted one after the other gives the amplitude for each frequency bin along the y-axis and the time step running along the x-axis. This gives us a



Layer	Support	Filt dim.	# filts.	Stride	Data size
conv1	7×7	1	96	2×2	254×148
mpool1	3×3	-	-	2×2	126×73
conv2	5×5	96	256	2×2	62×36
mpool2	3×3	-	-	2×2	30×17
conv3	3×3	256	256	1×1	30×17
conv4	3×3	256	256	1×1	30×17
conv5	3×3	256	256	1×1	30×17
mpool5	5×3	-	-	3×2	9×8
fc6	9×1	256	4096	1×1	1×8
apool6	1×n	-	-	1×1	1×1
fc7	1×1	4096	1024	1×1	1×1
fc8	1×1	1024	1251	1×1	1×1

Figure 2.1: CNN Architexture. [2] VoxCeleb: a large-scale speaker identification dataset

2-dimensional input feature running for variable length in the time domain for each audio sample.
All audio is first converted to single-channel, 16-bit streams at a 16kHz sampling rate for consistency. The standard MFCC extraction technique is used with 25ms width for hamming window, strides of 10ms and 1024 point-FFT. To deal with the variable length in time, we chop up the sample in 3 second length clips, which gives us a 512x300 matrix at the end.

2.3 Dataset

We are using the VoxCeleb dataset which contains 1251 celebrity videos on youtube. The dataset provides the start and end time in each video for every celebrity. Due to the huge size of dataset we only operated on 1164 celebrities. In the end we had 341,021 clips (512x300 matrix) for training and 10,000 clips (512*300 matrix) for testing.

2.4 Architecture

The closed set identification effectively means that we can interpret the problem as multi-class classification in which the speaker-id can be treated as different classes. This allows us to use the useful and powerful VGG-18 architecture with modifications related to final layer. The modifications pertain to the temporal dependency that is present in the matrices along the rows. This led to the modification of the fully connected layer 6 of dimension 9×8 to fully connected layer of dimension 9×1 and then an average pooling layer of dimension 1×8. The intuition behind this is that the features extracted till the max pool layer 5 has enough context and cross frequency information, which can now be compressed along the different frequencies for each time step and later can be combined across time using a normal average pooling. This allows for temporal independence but not across frequencies. Using normal VGG architecture, we would have not distinguished between frequency and time dimension of data and would have assumed translation invariance accross frequency as well as time which is definitely not the right way. We only need to assume translation invariance across time, hence the modification to the final layer. Also, the modification reduces the number of parameters to be learned by almost a factor of 4 and helps in reducing the training time.

3 Implementation Details

3.1 Dataset Preparation

First, downloading the dataset was quite slow beacause there are multiple small files. So we downloaded and extracted POI's voice from the mp3 in parallel and concatenated the extracted voice from each video into one mp3. Next, we tried converting them into wav files but were too huge. So, we directly extracted MFCC features as directed in the paper from mp3 files and split them in 3 sec intervals (SR - 16000, MFCC 3 sec worth - 512 x 300) and stored these matrices in compressed numpy format. For this we used the python library *librosa*. Finally, dataset resulted in 450 GB size.

3.2 Training

We used TensorFlow for implementing the VGG-18 architecture as stated in the paper on NVIDIA P-6000 GPU. We used Batch-Normalization after every convolutional layer to reduce the training time and simultaneously acting as a regularizer, ReLU as the activation function and cross-entropy as the loss function. We fed the model with batch size of 40 with the help of a threaded iterator to parallelly utilize training time for file I/O and trained it for 6 epochs using early stopping. To ensure stochasticity in the learning process, we randomly shuffle the entire dataset after every epoch. Each epoch took around 8 hours to finish. We performed the Adam gradient update with learning rate of 0.01.

4 Experiments

4.1 Experimental Setup

For training and testing of the network, we use the same number of speakers i.e. 1164. train set is of size, test set is of size something. we compute the top-1(%) accuracy on the 10,000 matrices held out in form of the test set. We also give these results on baselines setup before going on to the actual results.

4.2 Baselines

The results from the VoxCeleb paper about the GMM-UBM, I-vector/PLDA and one-vs-rest binary SVM classifier is stated for the baseline. The accuracy stated in the paper about their architecture and model also serves as the baseline.

4.3 Results

The accuracy of the model described in the VoxCeleb paper has almost 20% improvement over the I-vector approach. They use a average pooling layer for variable length test data. We only compare the accuracy of fixed size test data. They have mentioned that mean and variance normalization plays a significant role in the superiority of their network which gives a 10% boost in the accuracy. Our model does better than the model described in the paper though the test set differs. We have lesser number of speaker in our both train and test dataset 1164 compared to the 1251 given in paper. To actually compare the results we ran the models given by them on our test set. The results validate that our model is better even though their training set contained a lot of our testing set utterances. We also saw that the mean and variance normalization which gave a huge boost to their model didn't matter much in our experiments. The mean and variance normalization only marginally improved the accuracy in our case.

Accuracy	Top-1(%)
I-vectors + SVM	49.0
I-vectors + PLDA + SVM	60.8
CNN-fc-3s(paper)	70.04
CNN-fc-3s no var. norm.	74.1
CNN-fc-3s	75.3

Results on VoxCeleb Dataset,
The I-vector results are on full dataset,
Remaining results are on reduced dataset

5 Conclusion

The accuracy improvement in case of CNNs clearly prove their effectiveness even in the domain of speech. With the improvement in the amount of speech dataset available, the networks can be trained more and we can achieve even more accuracy. Hence, the state of the art techniques like i-vectors prove to be more accurate when we have fewer amount of data and lesser computational resources but with the amount of data and compute resources increasing rapidly, CNNs and other deep learning methods prove to be more promising. However, there are certain disadvantages to this method. First, is the case of expanding the model to include more speakers. This requires changing the model architecture in CNNs, hence we require re-training of weights through the entire dataset just to add one speaker to the model. Another major disadvantage that it can potentially suffer from is the adverserial speech samples which can pose serious security threats.

References

[1] D. Garcia-Romero D. Snyder and D. Povey. Time delay deep neural network-based universal background models for speaker recognition. 2015.

[2] A. Nagrani, J. S. Chung, and A. Zisserman. Voxceleb: a large-scale speaker identification dataset. In *INTERSPEECH*, 2017.