Voice Speech Commands

Sherif Tohamy

***Abstract*— Automatic classification of sound commands is becoming increasingly important, especially for mobile and embedded devices. Many of these devices contain both cameras and microphones, and companies that develop them would like to use the same technology for both of these classification tasks. One way of achieving this is to represent sound commands as images, and use convolutional neural networks when classifying images as well as sounds. In this report we consider another approach to the problem of sound classification. Here we show representation of sounds (Wave frames, Spectrograms) and apply two different convolutional neural networks (CNNs) architectures in order to get the best performance although having small dataset with only (8000) samples. As a result we achieved a good classification accuracy by replacing the the simple CNN layers with ResNet50 arhciteture and applying a different learning technique by using two different optimizers.**

I. INTRODUCTION

Automatic speech recognition (ASR) is the art and science of having machine to identify sounds [1]. These sounds could be much beyond than speech and music. Among others it includes such examples as barking dogs, breaking glasses, crying babies and etc. Sound recognition is a key strategic technology that will be embedded in most connected devices offering AI capabilities. For example, every one of us has come across smartphones with mobile assistants such as Apple Siri, Amazon Alexa or Google Assistant. These applications are dominating and in a way invading human interactions. In the nearest future we will see exponential growth of speech recognition embedded devices that will assist to our every day lives. It requires to develop and optimize sound recognition algorithms that need to be fast enough to work in real time and support many different embedded platforms [2].

First, to consider the overall research domain, it would be useful to clarify what is encompassed by the term speaker recognition, which consists of two alternative tasks: speaker identification and verification. In speaker identification, the task is to identify an unknown speaker from a set of known speakers. Whether, in speaker verification, an unknown speaker claims an identity, and the task is to verify if this claim is true. This essentially comes down to comparing two speech samples/utterances and deciding if they are spoken by the same speakers.

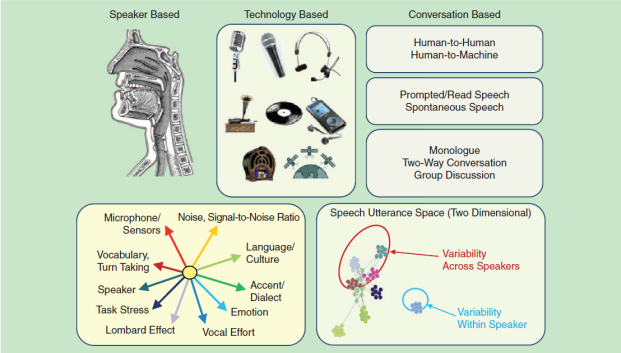

Fig. 1. Sources of variability in speaker recognition.

Unlike other forms of biometrics (e.g., fingerprints, irises, facial features, gait, and hand geometry) [3], human speech is a performance biometric. This makes speech signals prone to a large degree of variability. It is important to note that even the same person does not say the same words in exactly the same way every time (this is known as style shifting or intra speaker variability) [4]. To consider variability, (Figure 1) highlights a range of factors that can contribute to mismatch for speaker recognition. These can be partitioned based on three broad classes: 1) speaker based, 2) conversation based, and 3) technology based. Also, variability for speakers can be within speakers and across speakers.

The research community is largely driven by standardized tasks set forth by NIST through the speaker-recognition evaluation (SRE) campaigns [5]–[8]. A simple block diagram representation of an automatic speaker-verification system is shown

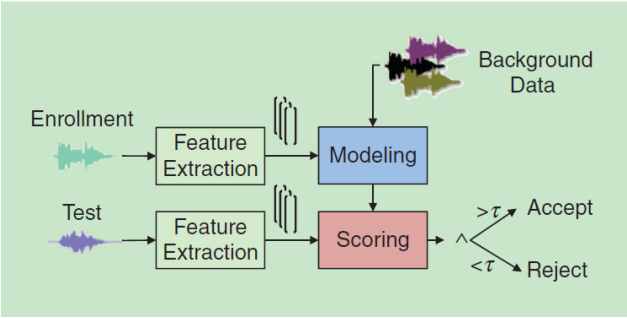

Fig. 2. An overall block diagram of a basic speaker-verification system.

in (Figure 2). However, in some automatic systems, the feature-extraction processes may be dependent on other speech utterances spoken by a diverse speaker population, as well as the enrollment speaker [9]. In short, the recent techniques make use of the general properties of human speech by observing many different speech recordings to make effective speaker-verification decisions. This is also intuitive, since we also learn how human speech varies across conditions over time. For example, if we only heard one language in our entire life, we would have difficulty distinguishing people speaking a different language [10].

A. ASR Approaches

Classical approaches in sound recognition system are based on understanding the components of human speech. A phoneme is a distinct unit in the sound system that helps to distinguish between meanings of words from a set of similar sounds corresponding to it pronounced in one or more ways. For example the word speech has the four phonemes: [11]. To find phonemes, speech signals are slowly timed where their characteristics are stationary over a short period of time. In the feature extraction step, acoustic observations are extracted in frames of typically 25 ms. For the acoustic samples in that frame, a multi-dimensional vector is calculated and on that vector a fast Fourier transformation is performed to transform a function of time.

Recently, deep learning-based approaches demonstrated performance improvements over conventional machine learning methods for many different applications [12]. Neural networks like LSTMs have taken over the field of Natural Language Processing [13]. CNNs have been applied to acoustic modeling before, notably by [14] and [15], in which convolution was applied over windows of acoustic frames that overlap in time in order to learn more stable acoustic features for classes such as phone, speaker and gender.

In this report we consider the CNN approach to the problem of sound classification. Here we show representation of sounds (Wave frames, Spectrograms) and apply convolutional neural networks (CNNs) in order to get the best performance although having small dataset with only (8000) samples.

II. Methods

A. ResNet

The Residual Neural Network (ResNet) was developed in 2015 by a group from the Microsoft research team [16]. They introduced a novel residual module architecture with skip connections. The network also features heavy batch normalization for the hidden layers. This technique allowed the team to train very deep neural networks with 50, 101, and 152 weight layers while still having lower complexity than smaller networks like VGGNet (19 layers). ResNet was able to achieve a top-5 error rate of 3.57% in the ILSVRC 2015 competition, which beat the performance of all prior ConvNets. To solve the vanishing gradient problem, the authors of ResNet created a shortcut that allows the gradient to be directly back propagated to earlier layers. These shortcuts are called skip connections: they are used to flow information from earlier layers in the network to later layers, creating an alternate shortcut path for the gradient to flow through. Another important benefit of the skip connections is that they allow the model to learn an identity function, which ensures that the layer will perform at least as well as the previous layer. This combination of the skip connection and convolutional layers is called a residual block. Similar to the Inception network, ResNet is composed of a series of these residual block building blocks that are stacked on top of each other (figure 3). we have implemented ResNet50: a version of the ResNet architecture that contains 50 weight layers (hence the name).

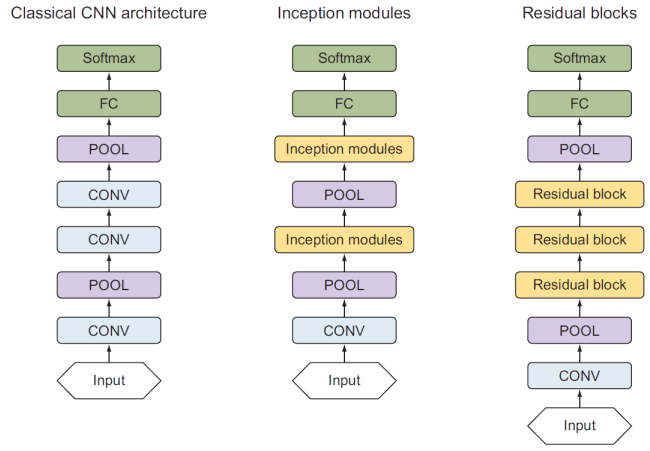

Fig. 3. Classical CNN architecture (left). The Inception network consists of a set of inception modules (middle). The residual network consists of a set of residual blocks (right).

Advanced CNN architectures with 18, 34, 101, and 152 layers by following are illustrated in (figure 4) from the original paper.

B. Learning in the CNN

First, the audio file will initially be read as a binary file, which needs to be converted into a numerical tensor. A WAV file contains time series data with a set number of samples per second. Each sample represents the amplitude of the audio signal at that specific time. In a 16-bit system, like the files in our dataset, the values range from -32768 to 32767. The sample rate for this dataset is 16kHz. Second, we convert the waveform into a spectrogram, which shows frequency changes over time and can be represented as a 2D image. This can be done by applying the short-time Fourier transform (STFT) to convert the audio into the time-frequency domain. A Fourier transform converts a signal to its component frequencies, but loses all time information. Instead, the STFT splits the signal into windows of time and runs a Fourier transform on each window, preserving some time information, and returning a 2D tensor that you can run standard convolutions on. STFT produces an array of complex numbers representing magnitude and phase.

For the first model (version 1), we use a simple convolutional neural network (CNN). The model also has the following additional preprocessing layers: 1) A Resizing layer to down sample the input to enable the model to train faster. 2) A Normalization layer to normalize each pixel in the image based on its mean and standard deviation. And for the second model (version 2), we changed the simple CNN architecture with ResNet50 and divied the learning process into two stages. In the first stage we used Adam optimizer [17], then in second stage we used stochastic gradient descent (SGD) which resulting in high test accuracy.

III. Experiments and Results

A. Dataset

Speech_commands [18] is an audio dataset of spoken words designed to help train and evaluate keyword spotting systems. Its primary goal is to provide a way to build and test small models that detect when a single word is spoken, from a set of ten target words, with as few false positives as possible from background noise or unrelated speech. We used portion of the Speech Commands dataset to classify a one second audio clip as ”down”, ”go”, ”left”, ”no”, ”right”, ”stop”, ”up” and ”yes” with 8000 WAV audio files. The original dataset consists of over 105,000 WAV audio files of people saying thirty different words.

B. Results

The two versions consumed the same time approximately for the training and validation process. We split the files into training, validation and test sets using a 90:5:5 ratio, respectively. For the first version we got 84% test accuracy and the model was somehow stable but has low test accuracy by comparing with the results of the second version 91%. By using a more deep CNNs (ResNet50) we have noticed that the deeper the network, the larger its learning capacity, and the better it extracts features from images. This mainly happens because very deep networks are able to represent very complex functions, which allows the network to learn features at many different levels of abstraction, from edges (at the lower layers) to very complex features (at the deeper layers).

IV. Discussion and Conclusion

In this report we proposed a CNN-based deep learning approach to solve the problem of sound classification. we apply two different convolutional neural networks (CNNs) architectures in order to

Layer name	Output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112x112	7x7, 64, stride 2				
conv2_x	56x56	3x3, maxpool, stride 2				
		[3x3, 64; 3x3, 64] x2	[3x3, 64; 3x3, 64] x3	[1x1, 64; 3x3, 64; 1x1, 256] x3	[1x1, 64; 3x3, 64; 1x1, 256] x3	[1x1, 64; 3x3, 64; 1x1, 256] x3
conv3_x	28x28	[3x3, 128; 3x3, 128] x2	[3x3, 128; 3x3, 128] x4	[1x1, 128; 3x3, 128; 1x1, 512] x3	[1x1, 128; 3x3, 128; 1x1, 512] x4	[1x1, 128; 3x3, 128; 1x1, 512] x8
conv4_x	14x14	[3x3, 256; 3x3, 256] x2	[3x3, 256; 3x3, 256] x6	[1x1, 256; 3x3, 256; 1x1, 1024] x3	[1x1, 256; 3x3, 256; 1x1, 1024] x23	[1x1, 256; 3x3, 256; 1x1, 1024] x36
conv5_x	7x7	[3x3, 512; 3x3, 512] x2	[3x3, 512; 3x3, 512] x3	[1x1, 512; 3x3, 512; 1x1, 2048] x3	[1x1, 512; 3x3, 512; 1x1, 2048] x3	[1x1, 512; 3x3, 512; 1x1, 2048] x3
	1x1	Average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Fig. 4. Architecture of several ResNet variations from the original paper.

get the best performance although having small dataset with only (8000) samples. As a result we achieved a good classification accuracy by replacing the the simple CNN layers with ResNet50 architecture and applying a different learning technique by using two different optimizers. We have proved that the deeper the network, the larger its learning capacity, and the better it extracts features from images by showing how our second version with ResNet50 outperformed the first version with simple CNN layers.

REFERENCES

[1] Dan Jurafsky and James H Martin. "Speech and language processing" volume 3. Pearson London, 2014.

[2] Roman A Solovyev, Alexandr A Kalinin, Alexander G Kustov, Dmitry V Telpukhov, and Vladimir S Ruhlov. "Fpga implementation of convolutional neural networks with xedpoint calculations." arXiv preprint arXiv:1808.09945, 2018

[3] [Online]. Available: www.biometrics.gov

[4] P. Eckert and J. R. Rickford, "Style and Sociolinguistic Variation". Cambridge, U.K.: Cambridge Univ. Press, 2001.

[5] A. F. Martin and M. A. Przybocki, "The NIST speaker recognition evaluations: 1996-2001," in Proc. Odyssey: The Speaker and Language Recognition Workshop, Crete, Greece, pp. 1–5, 2001.

[6] C. S. Greenberg and A. F. Martin, "NIST speaker recognition evaluations 1996-2008," in Proc. SPIE Defense, Security, and Sensing, 2009, pp. 732411–732411-12.

[7] A. F. Martin and C. S. Greenberg, "The NIST 2010 speaker recognition evaluation." in Proc. Interspeech, 2010, pp. 2726–2729.

[8] G. R. Doddington, M. A. Przybocki, A. F. Martin, and D. A. Reynolds, "The NIST speaker recognition evaluation–overview, methodology, systems, results, perspective," Speech Commun., vol. 31, no. 2–3, pp. 225–254, June 2000.

[9] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," IEEE Trans. Audio, Speech, Lang. Process., vol. 19, no. 4, pp. 788–798, 2011.

[10] T. K. Perrachione, S. N. Del Tufo, and J. D. Gabrieli, "Human voice recognition depends on language ability," Science, vol. 333, no. 6042, pp. 595–595, July 2011.

[11] Rainer E Gruhn, Wolfgang Minker, and Satoshi Nakamura. Statistical pronunciation modeling for non-native speech processing. Springer Science & Business Media, 2011.

[12] Yann LeCun, Yoshua Bengio, and Georey Hinton. Deep learning. nature, 521(7553):436, 2015.

[13] Klaus Gre, Rupesh K Srivastava, Jan Koutnk, Bas R Steunebrink, and Jurgen Schmidhuber. Lstm: A search space odyssey. IEEE transactions on neural networks and learning systems, 2017.

[14] H. Lee, P. Pham, Y. Largman, and A. Ng, "Unsupervised feature learning for audio classification using convolutional deep belief networks," in Proc. Adv. Neural Inf. Process. Syst. 22, 2009, pp. 1096–1104.

[15] D. Hau and K. Chen, "Exploring hierarchical speech representations using a deep convolutional neural network," in Proc. 11th UK Workshop Comput. Intell. (UKCI '11), Manchester, U.K., 2011.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep Residual Learning for Image Recognition," 2015,

http://arxiv.org/abs/1512.03385.
[17] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” ICLR, 2015.
[18] P. Warden, “Speech commands: A dataset for limited-vocabulary speech recognition,” 2018, arxiv:1804.03209