Title
Non-linguistic Vocalization Recognition Based on Convolutional, Long Short-Term Memory, Deep Neural Networks

Author
Qiu, Liang

Publication Date
2018

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA
Los Angeles

Non-linguistic Vocalization Recognition Based on Convolutional, Long Short-Term Memory, Deep Neural Networks

A thesis submitted in partial satisfaction
of the requirements for the degree
Master of Science in Electrical and Computer Engineering

by

Liang Qiu

2018

ABSTRACT OF THE THESIS

Non-linguistic Vocalization Recognition Based on Convolutional, Long Short-Term Memory, Deep Neural Networks

by

Liang Qiu
Master of Science in Electrical and Computer Engineering
University of California, Los Angeles, 2018
Professor Lei He, Chair

Non-linguistic Vocalization Recognition refers to the detection and classification of non-speech voice such as laughter, sneeze, cough, cry, screaming, etc. It could be seen as a subtask of Acoustic Event Detection (AED). Great progress has been made by previous research to increase the accuracy of AED. On the front end, multiple kinds of features such as Mel-Frequency Cepstral Coefficients (MFCCs), Gammatone Cepstral Coefficients (GTCCs) and many other hand-crafted features were explored. While on the back end, models or methods such as Gaussian Mixture Models (GMMs), Hidden Markov Models (HMMs), Bags-of-Audio-Words (BoAW), Support Vector Machine (SVM) and various types of neural networks were experimented.

Recent researches on Automatic Speech Recognition (ASR) and Acoustic Scene Classification (ASC) show the advantage of using Convolutional, Long Short-Term Memory, Deep Neural Networks (CLDNNs) on audio processing tasks. In this thesis, I am building a non-linguistic vocalization recognition system using CLDNNs. Log Mel-filterbank coefficients are adopted as input features and data augmentation methods such as random shifting and noise mixture are discussed. The built system is evaluated on a custom dataset collected from several resources and tested for real time application. The performance of CLDNNs for non-linguistic vocalization recognition is also compared with hybrid GMM-SVMs, Convolutional Neural Networks, Long Short-Term Memory and a fully connected Deep Neural

Network trained on VGGish embeddings.

The results indicate that CLDNNs outperform the other models in classification precision and recall. Visualization of CLDNNs are presented to help understand the framework. The model is proved accurate and fast enough for real time applications.

The thesis of Liang Qiu is approved.

Abeer A H Alwan

Song-Chun Zhu

Lei He, Committee Chair

University of California, Los Angeles

2018

To my parents, teachers and friends

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to thank my advisor, Prof. Lei He for his support and guidance throughout this research. This thesis would not have been possible without the efforts he put on me. I also want to express my appreciation to Prof. Abeer Alwan and students in UCLA Speech Processing and Auditory Perception Laboratory. They provided very valuable suggestions when I was doing this research. I want to thank Prof. Song-Chun Zhu for his significant suggestions on applying this research on practical applications. I want to thank my friends Wanyi Zhang, Sahil Jayaram, Nelson Solano, Ning Zhang, Nishant Shukla, Rui Fang. Last but not the least, I want to thank my family for their support.

CHAPTER 1

Introduction

1.1 Overview and Motivation

Non-linguistic Vocalization Recognition (NVR) is defined as the detection and analysis of non-speech human voice such as laughter, sneeze, yawn, etc, based on acoustic features. It can be seen as a subtask of Acoustic Event Detection (AED), which is defined as the recognition of any general individual sound events in audio, requiring estimation of onset and offset for distinct sound event instances and identification of the sound [MHV16].

Non-linguistic Vocalization Recognition have shown their application on multiple tasks such as emotion detection, audio segmentation, multimedia content retrieval, and acoustic surveillance [TGP16]. While speech is arguably one of the most important types, non-linguistic vocalization provides important information as well. For instance, screaming usually signifies scare and the emotion state can be labeled as negative. A response like "It's okay." might be made by a dialog manager. Laughter, on the other hand, can be labeled as positive. Response like "What's so funny?" might be appropriate. Additionally, coughing may indicate a negative status, and the temperature should be adjusted in a smart home environment [LL09]. Besides the emotion detection, NVR is also a necessary part of a hierarchical audio classification and recognition system. Such a system firstly classifies audio data into speech, non-speech, and environment sounds and NVR will further classify the non-speech signals into different categories. Under this scheme, the users of any Automatic Speech Recognition (ASR) services can reduce unnecessary data transmission by only streaming the speech audio data to the service providers, thus save them a lot of money each year. What's more, because NVR model doesn't need a lexicon model and a language model

for decoding, it's usually much smaller compared to a normal ASR system so that it could be deployed on device, working as a pre-filter for the downstream sound processing modules.

However, compared to the great progress researchers have made on speech recognition in the past sixty years, less work has been done on non-linguistic vocalization analysis. In light of the emerging deep neural networks powered by faster GPUs, modern ASR systems such as Google Assistant, Apple Siri or Amazon Alexa have very decent recognition accuracies especially in less noisy environment, but still none of them have the functionality to recognize non-speech vocalization right now. Part of the reasons is the lack of large, publicly available datasets. It prevents researchers from trying deep neural networks on this problem and the emergence of a more active community. Furthermore, distinguishing non-speech sound often requires analyzing an extended time period due to the lack of a clear sub-word unit [TGP16]. Unlike in spoken language, sound events are more random, both periodic and aperiodic, with less well defined occurrence patterns. These factors make the task of sound event detection and recognition inherently more difficult than ASR [Zha16]. So even though some features or ideas could be borrowed from normal speech recognition, this problem still needs further analysis and exploration. This thesis is aimed to: (1) start the process of building a large-scaled, fully-verified database of non-linguistic sounds, (2) introduce the implementation details and analysis of a NVR system based on Convolutional, Long Short-Term Memory, fully connected Neural Networks (CLDNNs).

1.2 Previous Work

In this section, I will introduce some previous work on non-linguistic vocalization recognition. Some of them are actually about acoustic event detection. I include them because NVR and AED almost share the same approaches. These previous researches vary wildly on the selection of acoustic features, the methods to aggregate frame-by-frame features and the classification models.

Traditional methods for NVR borrow techniques from normal ASR directly. For example, Mel-Frequency Cepstral Coefficients (MFCCs) were modeled with Gaussian Mixture Models

(GMMs) or Support Vector Machines or a fusion of both [Jan13] [TN06] [SID12]. There is also research trying to integrate such NVR into a large vocabulary ASR system by dividing non-speech voice into sub-word units [STB12]. But directly applying ASR methods to NVR task ignored the difference between speech and non-speech voice, thus led to inferior performance.

Researchers then turned to find more discriminative and robust features which are suitable for NVR task. Most of them were hand-crafted and derived from low-level descriptors such as MFCCs, filter banks or time-frequency descriptors. For instance, [VA12] found that, with a similar computational cost, the GTCCs are more effective than MFCCs in representing the spectral characteristics of non-speech signals, especially at low frequencies. [UBC12] proposed a new 2 dimensional feature set that can achieve a higher accuracy rate together with MFCCs than using MFCCs alone.

Before choosing an appropriate classification model, another big issue is how to aggregate these frame-by-frame descriptors to have a representation of the entire acoustic events that usually last for seconds. One way is to use Gaussian Mixture Models as I mentioned above. Another common method is the Bag of Audio Words (BoAW) approach [PA12], which is a mimic to the well established techniques for classifying text documents (bag-of-words) and image documents (bag-of-visual-words). During the training process of BoAW, all the training feature vectors are clustered (a common choice of the clustering algorithm is k-means) and the centroids of the resulting clusters are taken as the "words" to create a codebook. Then each feature vector can be replaced by a single index representing the nearest codeword to this vector (vector quantization). Then a Bag of Audio words is simply the histogram of all the codewords in a given file. These two aggregation methods, however, both discard the temporal order of the frame level features, causing considerable information loss [TGP16].

Within the past few years, inspired by the success of being applied to normal speech processing tasks, deep neural networks have also been introduced to this task and provide a new path to solve this problem. It seems when trained with sufficient data, we can rely on this powerful model to infer discriminative relationships from less refined but higher dimension-

ality input features. And plenty of work has achieved state-of-the-art robust performance when using higher dimensionality representations such as auditory images, spectrogram image features and spectrogram-derived sub-band power distribution along with deep neural networks. Also, using spectrogram image features makes it easy to directly borrow ideas from image processing area. For instance, [Zha16] used spectrogram image feature (SIF) and Concolutional Neural Networks (CNN). But considering most people don't have access to a large amount of data to train a neural network based on raw spectrogram image features, many neural network models are still based on MFCCs, which have become the de facto standard for audio parameterization.

There are so many tricks that we can play with when dealing with neural networks: topological structures, learning algorithms, data augmentation strategies, etc. Any of these factors can lead to a different behavior of neural networks and it still leaves an open problem how to find the optimal neural network for a specific problem. Here I introduce three representative works that I have found. [TGP16] introduced a Convolution Neural Network (CNN) with a large input field. They used a novel data augmentation method to introduce data variation and outperformed BoAW and classical CNNs. [Zha16], opposing to deep CNN architectures with multiple convolutional and pooling layers topped up with multiple fully connected layers, proposed a network consisting of only three layers: convolutional, pooling, and softmax layer. The varying-size convolutional filters at the convolution layer and 1-max pooling scheme at the pooling layer distinguishes it from other previous work. And [PHV16] used log Mel frequency bands as features and presented an approach to polyphonic sound event detection in real life recordings based on Bidirectional Long Short-Term Memory (BLSTM) Recurrent Neural Networks (RNNs) and achieved state-of-the-art performance in 2016.

1.3 Datasets

In this section, I will introduce some useful datasets for the NVR task. I will also talk about the custom dataset I collected from part of these public datasets and audios that I recorded.

1.3.1 AudioSet

AudioSet [GEF17] 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. AudioSet provides a common, realistic-scale evaluation task for audio event detection. 128 dimensional embedding vectors of each clip trained by VGGish model was also included when AudioSet was released. But the shortcoming of this dataset is that many of the audios are not pure and clean. The annotators only verified the presence of sounds they heard within YouTube segments and there is usually other kinds of noise or sounds overlapped on the labeled sound, which makes it hard to use.

1.3.2 Freesound Datasets

Freesound Database (FSD) [FPF17] includes a variety of everyday sounds, from human and animal sounds to music and sounds made by things, all under Creative Commons licenses. It's also organized following the AudioSet Ontology, but what makes it a better choice than AudioSet is that most of the audio data is clean and contains single events.

1.3.3 TUT Database

TUT Sound Events 2016 [MHV16] contains annotations for individual sound events, specially created for Sound Event Detection. High quality binaural audio was recorded, with an average duration of 3-5 minutes per recording, considering this is the most likely length that someone would record in everyday life. The equipment used for recording this specific dataset consists of binaural Soundman OKM II Klassik/studio A3 electret in-ear microphones and Roland Edirol R09 wave recorder using 44.1 kHz sampling rate and 24 bit resolution. Nouns were used to characterize each sound source, and verbs to characterize the sound production mechanism, whenever this was possible. Because of the overlapping sounds, each recording was listened multiple times to verify.

TUT Sound Events 2016 dataset consists of two common everyday environments: one outdoor situation (residential area) and one indoor situation (home). In residential area, classes include (object) banging, bird singing, car passing by, children shouting, people speaking, people walking, wind blowing; In home situation, classes include (object) rustling, (object) snapping, cupboard, cutlery, dishes, drawer, glass jingling, object impact, people walking, washing dishes, water tap running.

1.3.4 RWCP Sound Scene Database

The RWCP Sound Scene Database [NHA00] includes non-speech sounds recorded in an anechoic room, reconstructed signals in various rooms, impulse responses for a microphone array, speech data recorded with the same array, and recordings of background noises. It is intended for use when simulating sound scenes. It was developed by the Real Acoustic Environments Working Group of the Real World Computing Partnership (RWCP). The data was recorded from 1998 to 2000.

1.3.5 SSPNet Vocalization Corpus

The SSPNet Vocalization Corpus [SSB13] includes 2763 audio clips (11 seconds each) containing at least one laughter or filler instance. Overall, the corpus involves 120 subjects (63 females and 57 males). The clips are extracted from phone calls where two fully unacquainted speakers try to solve the Winter Survival Task.

1.3.6 Custom Dataset

Considering the lack of a large, open, fully-verified database for Non-linguistic Vocalization Recognition and Acoustic Event Detection, we started to build our own dataset containing a number of sound classes organized by AudioSet Ontology. The data is stored in a MongoDB database and a web based interface was created to view and manage the database content. Starting from human voice in household situation, we collect data from online resources mentioned above and also record some ourselves. For now, we have 1,000 clips (1 second

each) for three classes: laughter, sneeze and speech. Most of them are from AudioSet and Freesound Datasets. Each clip has been listened and verified the presence of the labeled class. We made sure two classes of sounds wouldn't appear in a single clip. I also collected some noise data under different conditions such as doing the dishes, cart miaow, biking, water tap running, pink noise and white noise. During training, the noise data will be randomly mixed with the clean data to make the NVR system more robust in practice.

1.4 Thesis Outline

The rest of this thesis is organized as follows.

Chapter 2 describes the workflow and implementation details of the Non-linguistic Vocalization System based on CLDNNs.

Chapter 3 presents the comparison of CLDNNs and other popular models for NVR tasks such as GMM-SVMs, CNN, LSTM and a DNN model trained on VGGish embeddings.

Chapter 4 summarizes the key results and provide directions for future work.

CHAPTER 2

System Workflow and Implementation

2.1 Acoustic Features

Feature extraction relying on signal processing techniques plays an important role in speech/non-speech related systems. The feature selection usually depends on the targeted tasks and appropriate features will significantly simplify the training process of downstream classifiers and improve the final performance. Mel-Frequency Cepstral Coefficients (MFCCs) were de facto standard features for a long time. But more recently, in light of the emergence of deep neural networks, filter banks are becoming increasingly popular. Computing filter banks and MFCCs involve almost the same procedure. I will illustrate this procedure step by step while including some of my implementation details.

The first step is to let the signal go through a pre-emphasis filter to amplify the high frequencies, which usually have smaller magnitudes compared to lower frequencies. The pre-emphasis filter can be a first-order filter described by the following equation:

$$y(t) = x(t) - \alpha x(t-1)$$

where a typical value for α is 0.97.

Then, we frame the signal into short-time signals. We do this based on the assumption that frequencies in a signal are stationary over a very short period of time. In my system, I chose a frame size of 25 ms and a stride of 10 ms which are typical choices in many ASR systems. Afterwards, a Hamming window function is applied to each frame in order to reduce spectral leakage. A hamming window has the following form:

$$w[n] = 0.54 - 0.46cos(\frac{2\pi n}{N-1})$$

where, $0 \leq n \leq N-1$, N is the window length.

Subsequently, we do an N-point Short-Time Fourier Transform (STFT) on each frame to get the magnitude spectrogram of the original signal. I selected an N of 512 to meet the requirement that N should not be smaller than the frame size (400 samples when the sampling rate is 16 kHz) to avoid time aliasing. The power spectrum can also be computed using the following equation:

$$P = \frac{|STFT(x_i)|^2}{N}$$

where, x_i is the i^{th} frame of signal x. We can either use a magnitude spectrogram or a power spectrogram; each has its pros and cons. In my case, I used magnitude spectrograms.

Finally, we can compute the filter banks by applying triangular filters on the Mel-scale magnitude spectrogram. The Mel-scale is a common re-weighting of the frequency dimension. It results in a lower dimensional and more perceptually-relevant representation of the audio by being more discriminative at lower frequencies and less discriminative at higher frequencies. Conversion from Hertz (f) to Mel (m) can use the following equations:

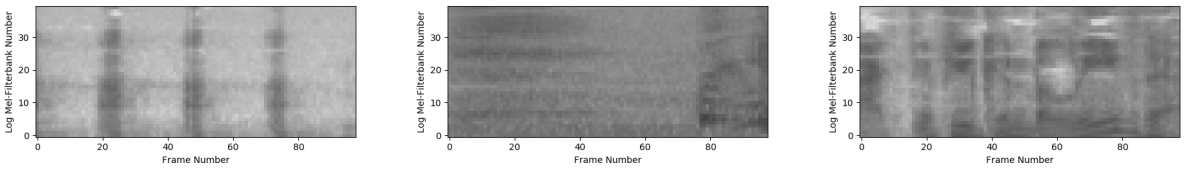
$$m = 2595 log_{10}(1 + \frac{f}{700})$$

It's also a common practice to compress the Mel-filterbank coefficients using logarithm, to balance the importance of detail in low and high energy regions.

It turns out that the log-Mel filterbank coefficients are usually highly correlated, which prevented it from being directly used by some machine learning algorithms. One solution is applying Discrete Cosine Transform (DCT) to decorrelate the coefficients and that leads to the Mel-Frequency Cepstral Coefficients (MFCCs). MFCCs, along with Gaussian Mixture Models-Hidden Markov Models became the standard method for many speech tasks for years. However, DCT seems an unnecessary operation for two reasons. First, it's a linear transformation, which would discard some information in highly non-linear speech signals. Second, applying DCT is only motivated by the limitation of some machine learning algorithms, while all the steps to calculate the log Mel-filterbank coefficients are motivated by the nature of speech signal or human perception characteristics. So when deep neural networks

become more and more mature recently, its robustness to highly correlated input allow us to use the log Mel-filterbank coefficients directly.

Somebody may even question whether the Fourier Transform is necessary. Since we assume the signal to be stationary within a very short time window, the Fourier transform would not discard too much information. Also, training classifiers on raw signals in the time domain usually requires more data to achieve a desired better performance. Considering my training data is limited, I chose 64 log Mel-filterbank coefficients as the acoustic features for CLDNNs and compared with a hybrid GMM-SVM classifier using 13 dimensional MFCCs. Example log Mel-spectrograms of laughter, sneeze and speech are shown in Figure 2.1.



(a) Laughter (b) Sneeze (c) Speech

Figure 2.1: Example log Mel-spectrograms

2.2 Convolutional, Long Short-Term Memory, Fully Connected Deep Neural Networks

Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM) and Deep Neural Networks (DNNs) are complementary in their modeling capabilities, as CNNs are good at reducing frequency variations, LSTMs are good at temporal modeling, and DNNs are appropriate for mapping features to a more separable space [SVS15]. Some previous works have shown the superiority of CLDNNs on Acoustic Scene Classification [GXL17]. In this project, I apply CLDNNs to Non-linguistic Vocalization Recognition and compare its performance with other popular models. The overall architecture of the CLDNN I implemented

is shown at Figure 2.2.

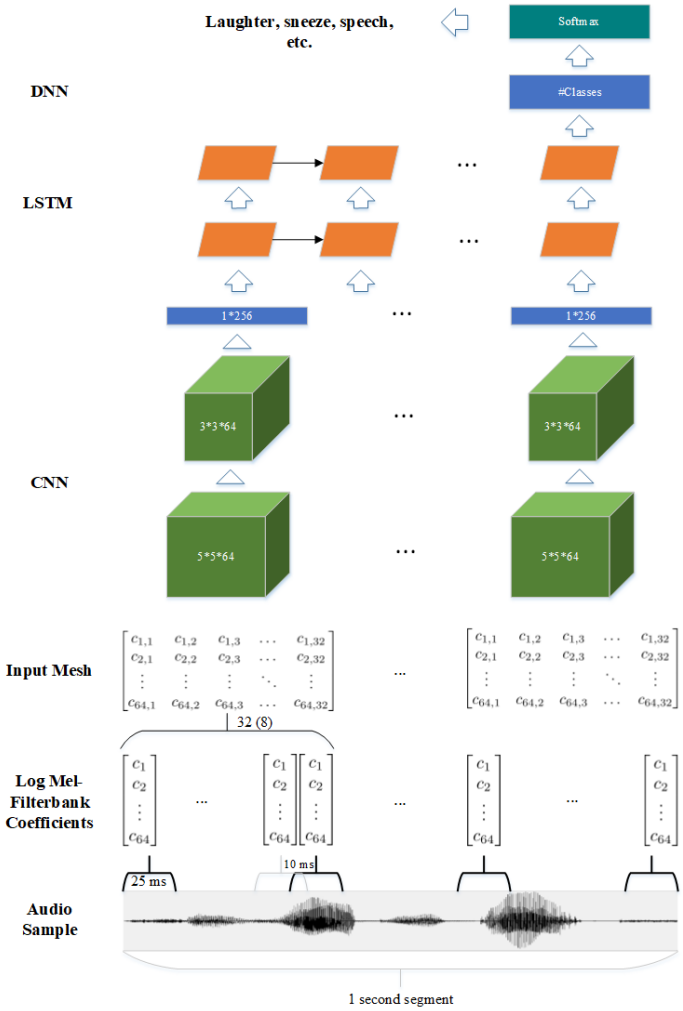


Figure 2.2: The implemented CLDNN framework

2.2.1 Input Layer

To utilize the contextual information around short time analysis frames and generate 2-D feature maps for CNN layers, I put 32 frames together into an input mesh every 8 frames. Notice that this input mesh is different from the short time analysis frame (25 ms with 10 ms stride) used to calculate the STFT. Each input mesh is composed out of a 64 dimensional log Mel-filterbank coefficient vector of frame i concatenated with its 31 contextual vectors. Therefore for each 1 second segment (sampled at 16 kHz), we will have $(1000-25)/10+1 = 98$

frames, $(98-32)/8+1=9$ input meshes and totally generate 9 $64*32$ 2-D feature maps, which will be input into the network. The input layer is illustrated by Figure 2.3.

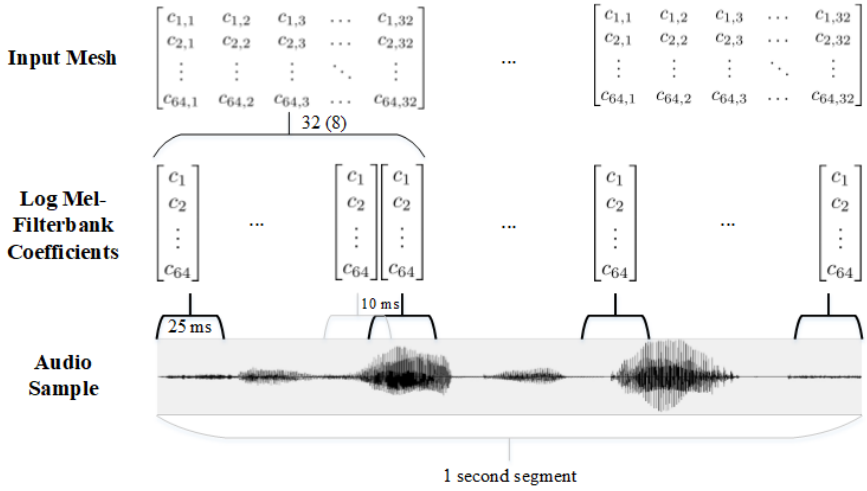


Figure 2.3: The input layer of CLDNN

And the magnitude spectrogram, log Mel-spectrogram and MFCCs of an example laughter waveform are presented at Figure 2.4, 2.5, 2.6, respectively.

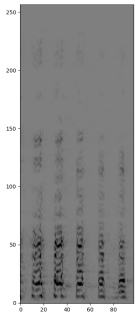


Figure 2.4: Magnitude spectrogram of laughter

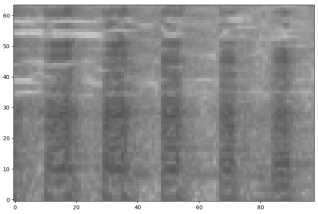


Figure 2.5: Log Mel-spectrogram of laughter

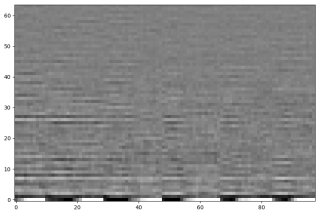


Figure 2.6: MFCCs of laughter

2.2.2 Convolutional Layer

To remove speaker variation and increase robustness, state-of-the-art systems usually adopt speaker adaptation techniques such as vocal tract length normalization (VTLN) or feature-

space maximum likelihood linear regression (fMLLR). In this project, we use a 2 layer convolutional neural network to do the similar work. Convolution neural networks, which could remove variation in the input feature mesh, are supposed to learn speaker adapted trained features. The mathematical expression of CNN is as follow: suppose we use an $m \times m$ filter ω, the output of the convolutional layer would be:

$$y_{ij}^{l} = ReLU(x_{ij}^{l})$$
$$x_{ij}^{l} = \sum_{a=0}^{m-1} \sum_{b=0}^{m-1} \omega_{ab} y_{(i+a)(j+b)}^{l-1}$$

where, l is the index of the layer.

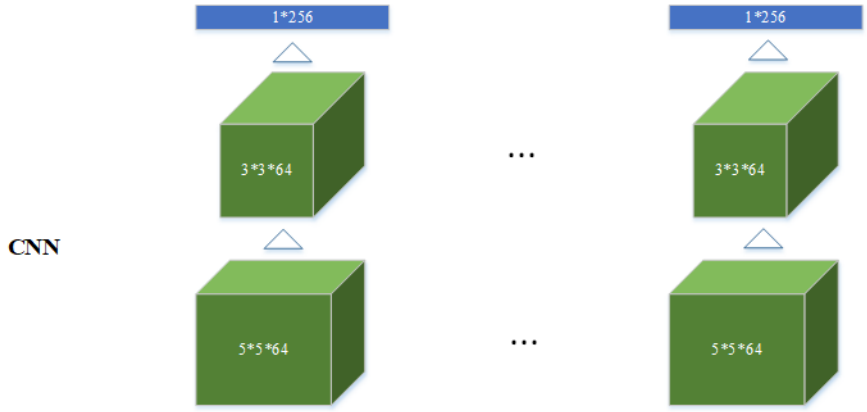


Figure 2.7: The convolutional layers of CLDNN

As shown in Figure 2.7, the first convolutional layer uses 64 filters of size 5*5, followed by a max pooling layer of size 2*2 and stride 2*2. The second layer uses 64 filters of size 3*3. All the strides for the convolutional layers are set to 1 and a dropout rate of 0.5 is used to prevent overfitting. The output of the second layer is then flattened and passed into a fully connected layer with 256 nodes, which could be seen as the embedding representation of the input mesh learned by CNN. The output of the first and second CNN layer are visualized in Figure 2.8 and Figure 2.9.

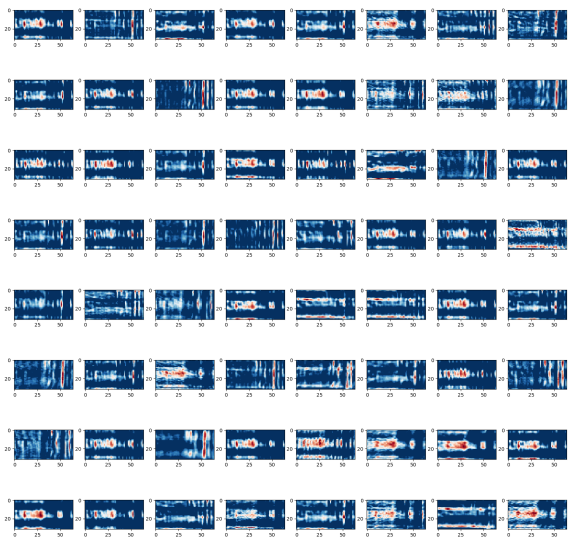


Figure 2.8: The visualization of the first convolutional layer

Figure 2.9: The visualization of the second convolutional layer

2.2.3 Long Short-Term Memory

After the frequency modeling is done by the CNN, the output of it is then passed to Long Short-Term Memory layers, which have good capability for temporal modeling. The mathematical expression of LSTM is as follow:

$$
\begin{aligned}
f_k &= \sigma(W_f \cdot [h_{k-1}, x_k] + b_f) \\
i_k &= \sigma(W_i \cdot [h_{k-1}, x_k] + b_i) \\
o_k &= \sigma(W_o \cdot [h_{k-1}, x_k] + b_o) \\
\widetilde{C_k} &= \tanh(W_c \cdot [h_{k-1}, x_k] + b_c) \\
C_k &= f_k * C_{k-1} + i_k * \widetilde{C_k} \\
h_k &= o_k * \tanh(C_k)
\end{aligned}
$$

where, C_k is the cell state, f_k is the forget gate that controls which information we are going to throw away from the last cell state, i_k is the input gate that decides which information we are going to store in the cell state, o_k is the output gate that decides what we are going to output.

For this project, I use 2 LSTM layers, where each layer has 256 hidden nodes, and they are illustrated by Figure 2.10.

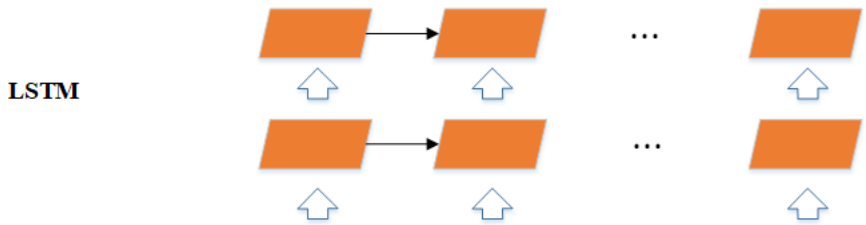


Figure 2.10: The LSTM layers of CLDNN

2.2.4 Fully Connected Layer

Finally, the output of the LSTM layers is passed to a fully connected layer, which transforms the tensors into a more separable space. In my project, I use one fully connected layer to

map the output of the LSTM layers to the non-linguistic vocalization classes. A softmax layer is also applied to calculate the probability distribution.

$$y_i^l = softmax(x_i^l)$$
$$x_i^l = \sum_j \omega_{ji} y_j^{l-1}$$

where, l is the index of the layer and i is the index of the non-linguistic vocalization class.

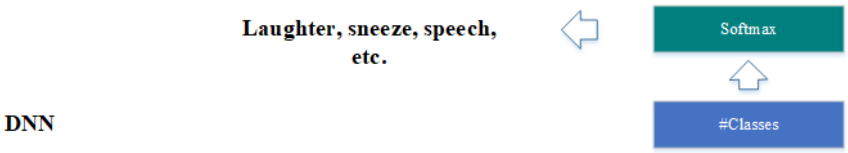


Figure 2.11: The fully connected layer of CLDNN

2.3 Data Preparation and CLDNN Training

2.3.1 Data Augmentation

To estimate how accurately the trained model will perform in practice, the collected dataset is divided into three sets: training set (80%), validation set (10%) and testing set (10%). Each audio segment is put into either of these three sets according to the hash value of its file name. This makes it less likely that validation or testing samples will be reused in training during the long run. I tried to put samples recorded under the same condition or from the same person into the same set so that the model would not learn from the environment or speaker information. Each set contains roughly the same proportions of class labels except silence and the details are reported in Table 2.1.

Also, in real situation, an acoustic event could occur at any point within the 1 second segment. So during training, I randomly shift the audio segment with a time range of 100 milliseconds and pad the shifted segment with zeros.

	Laughter	Sneeze	Speech	Silence	Total
Training	834	815	860	251	2760
Validation	98	115	91	31	335
Testing	111	113	93	33	350

Table 2.1: Number of samples of each class in three sets

2.3.2 Noise Robustness

To increase the noise robustness of the system, I also collected noise data under multiple conditions and mix them with clean training data with a random SNR from infinity to 3 dB. The statistics of the noise data are reported in Table 2.2.

Type	Length(s)
Doing the dishes	95
Cat miaowing	61
Biking	61
Running tap	60
Pink noise	61
White noise	60

Table 2.2: Noise data type and duration

2.3.3 Training Design

The project is implemented using Tensorflow framework. An Adam optimizer is used for the back propagation. The learning rate is set to 0.001 for the first 2000 steps and 0.0001 for the later 1000 steps. Every 400 steps, the model will be validated on the validation dataset and the learned weights are saved every 100 steps. The batch size is set to 50. I trained the model on a Dell XPS 8920 Desktop with Intel Core i7-7700 CPU and NVIDIA GeForce GTX 1080 GPU. The training and validation accuracy and cross entropy tracks are shown in

Figure 2.8 and Figure 2.9, where orange represents training and blue represents validation. From the figures, we can tell that with an Adam optimizer, the weights learning process converges quickly.

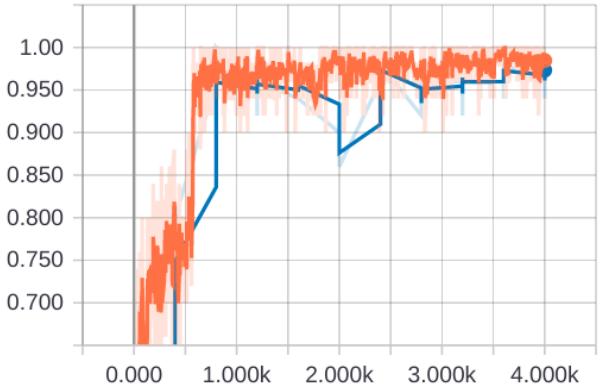


Figure 2.12: Training and validation accuracy tracks of CLDNN

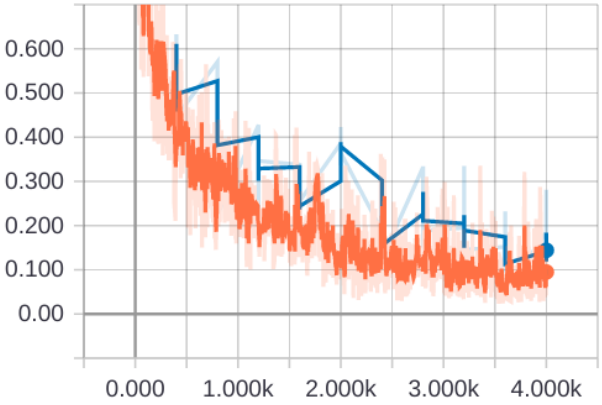


Figure 2.13: Training and validation cross entropy tracks of CLDNN

The histograms of some of the learned weights are shown in Figure 2.10. The final testing confusion matrix, precision, recall and F1 score are reported in Table 2.3 and Table 2.4.

2.4 Application in Real Time System

I also wrote a Remote Procedure Call (RPC) server with Apache Thrift, which is a set of code generation tools that allows developers to build RPC clients and servers that communicate seamlessly across programming languages. So I can run the trained model as a service and

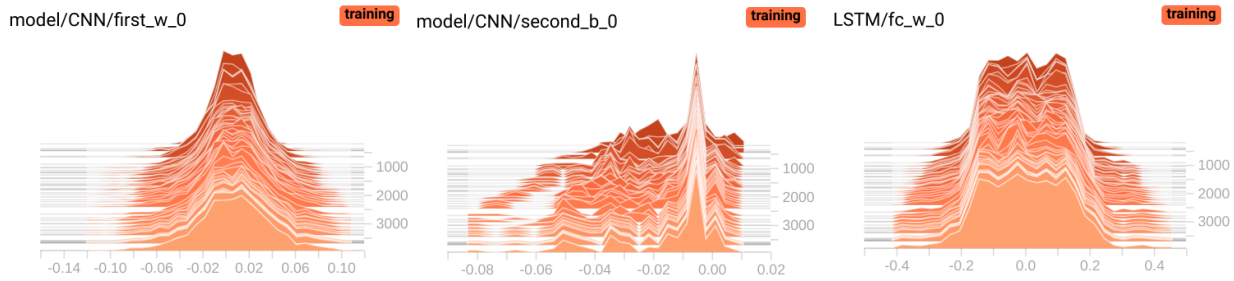


Figure 2.14: Histograms of some learned weights of CLDNNs

	Laughter	Sneeze	Speech	Silence
Laughter	99	11	1	0
Sneeze	1	111	1	0
Speech	0	3	90	0
Silence	0	0	0	33

Table 2.3: Final testing confusion matrix of CLDNN. Note that rows are true classes and columns are predicted classes.

	Precision	Recall	F1
Laughter	0.99	0.89	0.94
Sneeze	0.89	0.98	0.93
Speech	0.98	0.97	0.97
Silence	1	1	1
Average	0.97	0.96	0.96

Table 2.4: Final testing precision, recall and F1 score of CLDNN.

do inference in real time. I used an Android App to stream audio data to the server and estimate the inference latency. On the same Dell XPS 8920 Desktop, it takes approximately 8 milliseconds to classify a one-second clip. So we can conclude the latency is tolerable for a real time system in practical use.

2.5 Conclusion

In this chapter, I talked about the reason I chose log Mel-filterbank coefficients as the input acoustic features. With neural networks which are less susceptible to highly correlated features, the DCT operation seems an unnecessary step. I also introduced the concept and implementation details of the CLDNN framework. Model architecture and visualization are presented to help understand the framework. During the development, I found data augmentation, noise robustness and training design are of great significance. So I did a number of experiments with different settings and hyper-parameters, and proved the model's practicability in a real time system.

CHAPTER 3

Performance Analysis and Comparison

In the previous chapter, I talked about the implementation and training process of a CLDNN system for Non-linguistic Vocalization Recognition. To compare the performance of CLDNNs with other popular models and show its superiority, I also constructed hybrid Gaussian Mixture Models-Support Vector Machine (GMM-SVM), single type neural networks such as Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM) and Deep Neural Network with pre-trained VGGish embeddings. I will introduce them one by one and compare their performance with CLDNNs.

3.1 Hybrid Gaussian Mixture Models-Support Vector Machine

I first implemented a hybrid Gaussian Mixture models-Support Vector Machine [Jan13]. 13 dimensional MFCCs are first calculated using magnitudes of the Short-Time Fourier Transform with a window size of 25 milliseconds, a window stride of 10 milliseconds and a periodic Hamming window. So for each one second sample, we will have a 13*98 dimensional feature map. Each dimension of the MFCCs is then modeled using a Gaussian Mixture Model with N components. A supervector is thus constructed by stacking the mean and covariance values of the Gaussians. So with 13 dimensional MFCCs and if 6 Gaussians are used, 13*6*2=156-elements long supervectors will be created and the SVMs are supposed to operate in 156-dimensional space. I did a grid search of the component number, kernel type (linear, polynomial, radial basis function and sigmoid) and regularization parameter using scikit learn SVM package. In Table 3.1 and Table 3.2, I report the best training result with a component number of 6, using linear SVM kernel and a regularization (C) of 0.1.

	Laughter	Sneeze	Speech	Silence
Laughter	54	10	31	16
Sneeze	3	43	26	41
Speech	0	3	90	0
Silence	0	0	0	33

Table 3.1: Final testing confusion matrix of GMM-SVM. Note that rows are true classes and columns are predicted classes.

	Precision	Recall	F1
Laughter	0.95	0.49	0.64
Sneeze	0.77	0.38	0.51
Speech	0.61	0.97	0.75
Silence	0.37	1	0.54
Average	0.67	0.71	0.61

Table 3.2: Final testing precision, recall and F1 scores of GMM-SVM.

From the result, we can find that it's very likely for GMM-SVM to get confused between sneeze and silence. It's understandable since most part of the sneeze samples are silence and GMM-SVM doesn't have the capability to model the temporal information.

3.2 Single Type Neural Networks

As I talked in the last chapter, CNNs have good frequency variation modeling capability while LSTMs provide good temporal modeling. To prove their complementarity, I also implemented two single-type neural networks, Convolutional Neural Network and Long Short-Term Memory for the NVR task respectively. The inputs to the CNN and the LSTM are both the 64*98 dimensional log Mel-spectrograms.

3.2.1 Convolutional Neural Network

In light of the success of CNN in image processing, many researchers have tried to prove its efficiency in Acoustic Event Detection [ZMS15] [TGP16]. The constructed CNN has two convolutional layers. The first layer has 32 filters with size of 5*5 and stride 1*1, followed by a max pooling layer of size 2*2 and stride 2*2. The second layer has 64 filters with size of 3*3 and stride 1*1 as well.

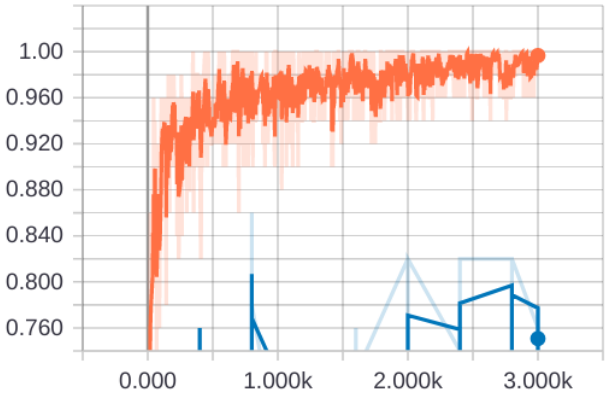


Figure 3.1: Training and validation accuracy tracks of CNN.

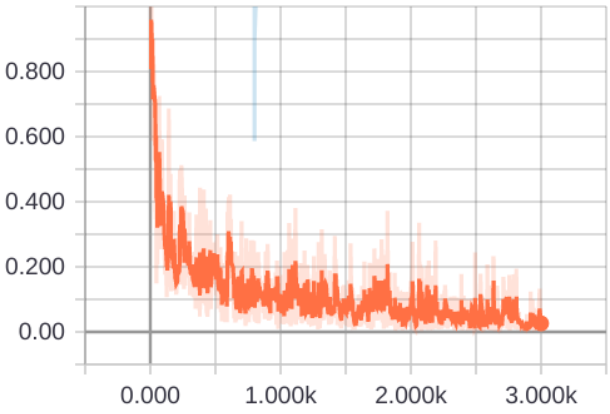


Figure 3.2: Training and validation cross entropy tracks of CNN.

A dropout rate of 0.5 is used to prevent overfitting and the activation function is ReLU. The final fully connected layer has 256 hidden nodes. The training and validation accuracy and cross validation tracks are shown as Figure 3.1 and Figure 3.2 respectively, where orange represents training and blue represents validation.

From the tracking logs, we can see CNN seems getting overfitting on the training set but can't achieve the same accuracy on the validation set. The final testing confusion matrix and the precision/recall/F1 scores are reported in Table 3.3 and Table 3.4. We can see that though CNN is better than GMM-SVM, it's still wore than CLDNN.

	Laughter	Sneeze	Speech	Silence
Laughter	37	52	22	0
Sneeze	0	110	3	0
Speech	0	11	82	0
Silence	0	0	0	33

Table 3.3: Final testing confusion matrix of CNN. Note that rows are true classes and columns are predicted classes.

	Precision	Recall	F1
Laughter	1	0.33	0.5
Sneeze	0.64	0.97	0.77
Speech	0.77	0.88	0.82
Silence	1	1	1
Average	0.85	0.80	0.77

Table 3.4: Final testing precision, recall and F1 scores of CNN.

3.2.2 Long Short-Term Memory Network

Recurrent Neural Networks (RNNs) are good at modeling the temporal information that is naturally present in audio. Long Short-Term Memory, one variant of RNN, are capable to handle the "long-term dependencies" problem of normal RNNs. In this experiment, 3 layer Long Short-Term Memory with 64 nodes each are constructed. A dropout rate of 0.5 is used to prevent overfitting. And the last step output of the LSTM is mapped directly to the non-linguistic vocalization classes using a fully connected layer. The training and validation

accuracy and cross validation tracks are shown as Figure 3.3 and Figure 3.4 respectively.

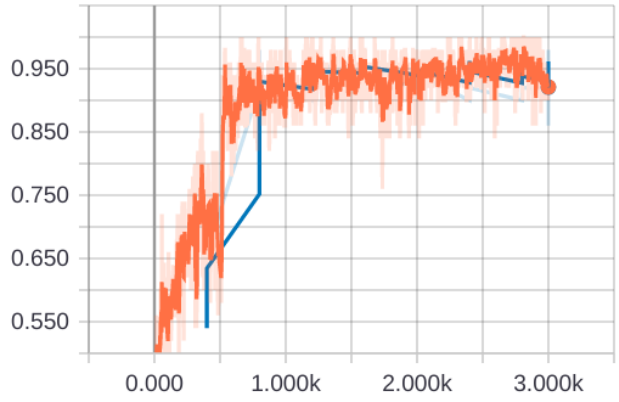


Figure 3.3: Training and validation accuracy tracks of LSTM.

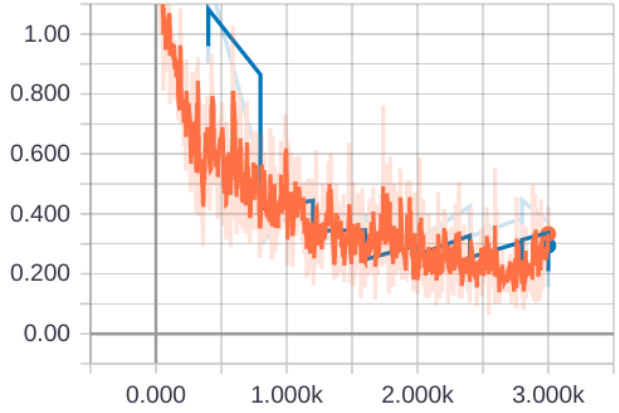


Figure 3.4: Training and validation cross entropy tracks of LSTM.

	Laughter	Sneeze	Speech	Silence
Laughter	89	15	0	7
Sneeze	0	110	3	0
Speech	0	2	91	0
Silence	0	0	0	33

Table 3.5: Final testing confusion matrix of LSTM. Note that rows are true classes and columns are predicted classes.

	Precision	Recall	F1
Laughter	1	0.80	0.89
Sneeze	0.87	0.97	0.92
Speech	0.97	0.98	0.97
Silence	0.83	1	0.90
Average	0.91	0.94	0.92

Table 3.6: Final testing precision, recall and F1 scores of LSTM.

The testing confusion matrix and the precision/recall/F1 scores are reported in Table 3.5 and Table 3.6. We can find that though it's still worse than CLDNN, LSTM can actually achieve the closest performance than other models being compared.

3.3 VGGish

VGGish is a variant of the VGG model released by Google's Sound Understanding team [HCE17]. The goal was to classify the soundtracks of a dataset of 70M training videos (5.24 million hours) with 30,871 video-level labels (YouTube-100M). Various CNN architectures such as fully connected Deep Neural Networks (DNNs), AlexNet, VGG, Inception, and ResNet were examined. They found that a model using embeddings from these classifiers did much better than raw features on the AudioSet [GEF17] Acoustic Event Detection classification task. So when releasing the initial AudioSet, they included 128-dimensional embeddings of each AudioSet segment produced from a VGG-like audio classification model called VGGish. Besides the VGGish definition, they also published supporting code to extract input features for the model from audio waveforms and to post-process the model embedding output into the same format as the released embedding features. For performance comparison with CLDNNs, I utilized these pre-trained embeddings as training data and input them into a DNN classifier for Non-linguistic Vocalization Recognition. The details of VGGish and my implemented model are as follow.

The feature extraction part is almost the same as what I did in CLDNNs. Each audio is

10 second long and re-sampled to 16 kHz. A log Mel-spectrogram is then calculated using magnitudes of the Short-Time Fourier Transform with a window size of 25 milliseconds, a window stride of 10 milliseconds, a periodic Hanning window and 64 Mel bins covering the range 125-7500 Hz. The log Mel-spectrogram of the 10 second audio waveform is then framed into non-overlapping samples of 960 milliseconds, where each example covers 64*96 log Mel-filterbank coefficients. Each sample will inherit the labels from the 10-second parent waveform. This causes the problem that many of the individual samples might not be informative about the labels and I will talk about this later.

VGGish has four groups of convolutional layers and they are stacked together in order. The first group and second group both have a single convolution layer with 64 filters and 128 filters, respectively. The third group has 2 convolutional layers with 256 filters each. The forth group has 2 convolutional layers with 512 filters each. All these filters have a size of 2*2 and stride 1*1. Each group of convolutional layers is followed by a max pooling layer of size 2*2 and stride 1*1. The output of the last max pooling layer is then input into two fully connected layers with 4096 hidden nodes each. Subsequently, the tensors are mapped to 128-dimensional embedding vectors. Finally, these embeddings are post-processed by applying a Principle Component Analysis (PCA) transformation as well as quantization to 8 bits per embedding element. For Non-linguistic Vocalization Recognition, I put the embeddings into a fully connected layer and predict the sound classes. The overall architecture of the DNN model using VGGish embeddings is illustrated by Figure 3.5.

The fully connected layer is then trained on the released pre-trained embeddings of AudioSet segments. Notice that, in AudioSet, each audio is given multiple labels, which is known as polyphonic event detection. To be consistent with the previous settings and also simplify the problem, I randomly choose one of the labels for each segment. I use a batch size of 70 and train the model for 30000 epochs using an Adam optimizer with a learning rate of 0.001. During testing, for fair comparison, I use the released code to extract features from the same testing set used for CLDNNs. The testing result is shown by Table 3.7 and Table 3.8.

It's not surprising that training using AudioSet VGGish embeddings doesn't provide

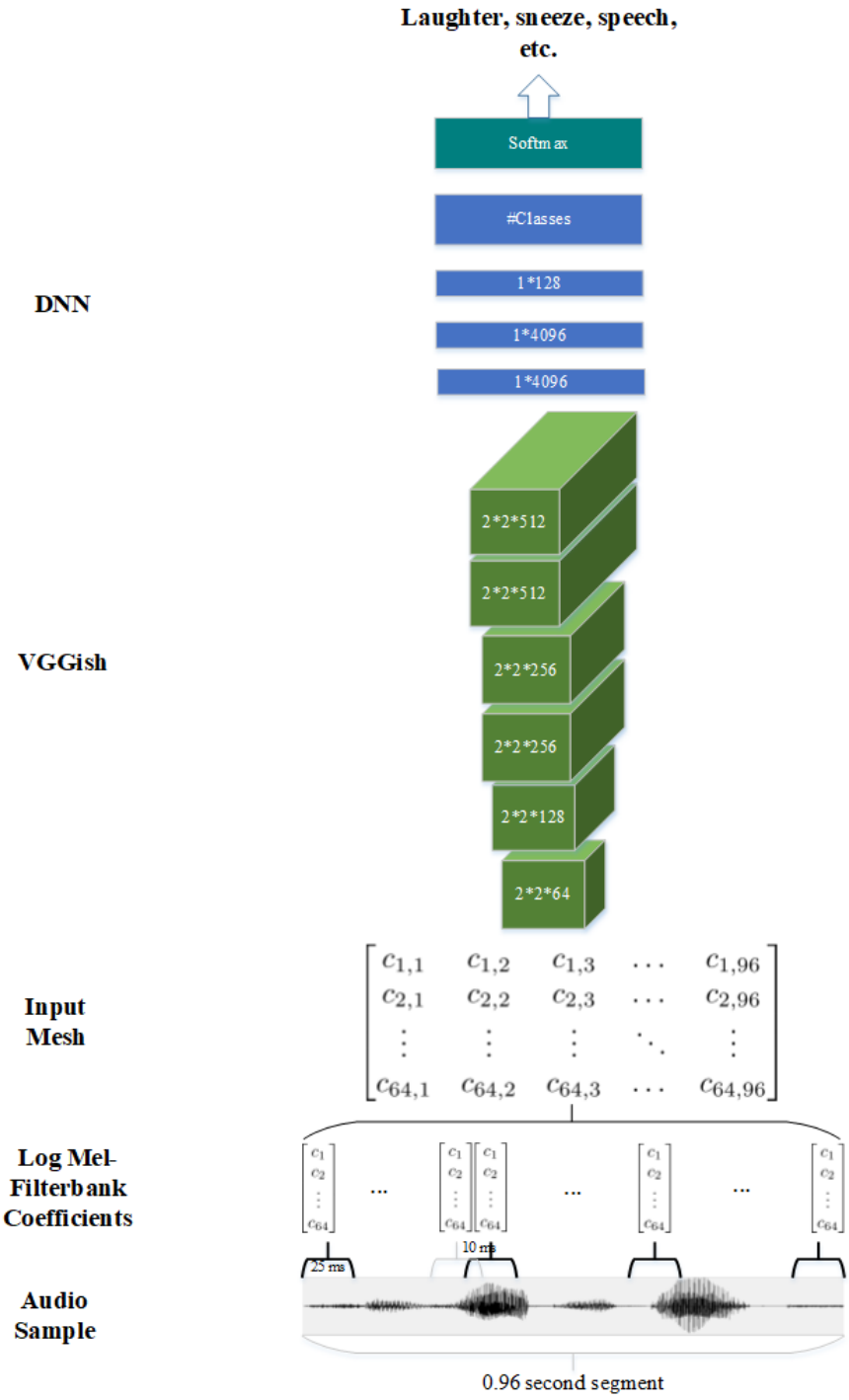


Figure 3.5: The VGGish-DNN framework.

a good result. First, it turns into a transfer learning problem when we train the model on AudioSet embeddings and test on my own dataset. Second, in AudioSet, audio waveforms are labeled when that sound class is proved presence by the annotators. But it's not guaranteed

	Laughter	Sneeze	Speech	Silence
Laughter	42	15	44	10
Sneeze	3	77	3	30
Speech	16	3	70	4
Silence	0	7	0	26

Table 3.7: Final testing confusion matrix of VGGish. Note that rows are true classes and columns are predicted classes.

	Precision	Recall	F1
Laughter	0.69	0.39	0.49
Sneeze	0.75	0.68	0.72
Speech	0.60	0.75	0.67
Silence	0.37	0.79	0.50
Average	0.60	0.65	0.59

Table 3.8: Final testing precision, recall and F1 of VGGish.

that each segment has only one class of sound and in most cases multiple kinds of sounds overlap. As I mentioned above, each of the 10 second audio is divided into non-overlapping 960 ms frames and each frame inherits the label of its parent video. This obviously will result in that many of the individual segments are actually uninformative about the labels. For example, for a 10 second segment labeled as laughter, several of the 10 samples framed from the segment might be silence, speech or any other sound classes. Even within the sample when laughter actually occurs, background noise such as people chatting with each other or cars passing by can appear and we have no control over it.

Of course, we can borrow the idea from the usage of GloVe word embeddings in Natural Language Processing. VGGish weights will only be used for initialization and they can be trained together with the downstream classifier on raw AudioSet data. However, that limits our model must have the same topological structure as the VGGish at the bottom

and we need to meet the constraint that each sample should be 0.96 second long and the embedding size should be 128. To customize or modify the model a little bit, we must retrain it on YouTube-100M which could take weeks on several GPUs. Also for me, instead of using a large scale but crude dataset like AudioSet, I prefer building a dataset from clean, fully verified audio data and mix them with noise data when I want to increase the noise robustness.

3.4 Conclusion

In this chapter, I compared the performance of CLDNN with other popular models and proved the superiority of CLDNN on the NVR task. Among the models I constructed, SVM-GMM takes shortest time to train but it requires all the training data be read into the memory. While for neural networks, training data can be fed using mini batches. What's more, GMM-SVM only provides an average precision of 0.67 and an average recall of 0.71, which is far from practical usage. Compared with single-type neural networks, LSTMs achieve the closest performance as CLDNN. It makes sense because of LSTM's capability to model temporal information. The improvement of CLDNN over LSTMs also prove the complementarity between CNNs and LSTMs. I also tried to train a DNN model on the embeddings pre-trained using VGGish on a large scale dataset for soundtrack classification. But the result was not good because of the existence of noise and multiple kinds of sounds in the data samples.

CHAPTER 4

Summary and Future Work

4.1 Conclusions

This thesis extends the previous research on Non-linguistic Vocalization Recognition and Acoustic Event Detection. CLDNN, a combination of three types of neural networks, was constructed and evaluated on a dataset with samples from various resources.

Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM) and Deep Neural Networks (DNNs) are shown complementary in their modeling capabilities. According to my experiments, the CLDNN classifier got very high precision and recall on the collected dataset after a few thousand steps' training. Multiple data augmentation techniques were applied to improve the model robustness. To test the model's practical usage, it was also run as part of a real time system. In general, the non-linguistic vocalization classifier based on CLDNN is accurate and fast.

Comparison between CLDDN and other popular models showed the superiority of CLDNN. GMM-SVM is easy to train but it easily get confused between sneeze and silence because it has no capability to model temporal information. LSTM did fairly well and achieved the closest performance as CLDNN. But with CNN, CLDNN can rely on it to learn speaker adapted features, thus remove the feature variance. A DNN classifier using pretrained VGGish embeddings of AudioSet was also tested but the result was not satisfying. It might be a better choice to build a new, well organized and fully verified dataset for acoustic sounds.

4.2 Future work

Now we have only collected data for three classes: laughter, sneeze and speech. To prove the model's capability, we are collecting more classes of data including human voice such as crying, sigh and non-human sounds such as dog barking, door bell or glass breaking. This classifier can then be integrated into a large vocabulary automatic speech recognition system. Also, by further integrating it with a backend dialog manager, it can make a chatting robot more intelligent and human like.

The classifier is currently running on a desktop server. I am planning to deploy the model on embedded devices such as a smart phone or a Raspberry Pi. By working offline, it will eliminate the latency caused by the network transmission and the classifier can work as a pre-filter prior to the speech recognition system to avoid unnecessary non-speech data streaming.

REFERENCES

[FPF17] Eduardo Fonseca, Jordi Pons, Xavier Favory, Frederic Font, Dmitry Bogdanov, Andres Ferraro, Sergio Oramas, Alastair Porter, and Xavier Serra. "Freesound Datasets: a platform for the creation of open audio datasets." *ISMIR, International Society for Music Information Retrieval Conference*, pp. 486–493, 2017.

[GEF17] Jort F. Gemmeke, Daniel P.W. 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." *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pp. 776–780, 2017.

[GXL17] Jinxi Guo, Ning Xu, Li Jia Li, and Abeer Alwan. "Attention based CLDNNs for short-duration acoustic scene classification." *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, pp. 469–473, 2017.

[HCE17] Shawn Hershey, Sourish Chaudhuri, Daniel P.W. Ellis, Jort F. Gemmeke, Aren Jansen, R. Channing Moore, Manoj Plakal, Devin Platt, Rif A. Saurous, Bryan Seybold, Malcolm Slaney, Ron J. Weiss, and Kevin Wilson. "CNN architectures for large-scale audio classification." *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pp. 131–135, 2017.

[Jan13] Artur Janicki. "Non-linguistic vocalisation recognition based on hybrid GMM-SVM approach." *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, pp. 153–157, 2013.

[LL09] Wen-Hung Liao and Yu-Kai Lin. "Classification of non-speech human sounds." *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, pp. 2695–2700, 2009.

[MHV16] Annamaria Mesaros, Toni Heittola, and Tuomas Virtanen. "TUT Database for acoustic scene classification and sound event detection." 2016.

[NHA00] Satoshi Nakamura, Kazuo Hiyane, Futoshi Asano, Takanobu Nishiura, and Takeshi Yamada. "Acoustical sound database in real environments for sound scene understanding and hands-free speech recognition." 2000.

[PA12] Stephanie Pancoast and Murat Akbacak. "Bag-of-Audio-Words approach for multimedia event classification." *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, pp. 1–4, 2012.

[PHV16] Giambattista Parascandolo, Heikki Huttunen, and Tuomas Virtanen. "Recurrent neural networks for polyphonic sound event detection in real life recordings." *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pp. 6440–6444, 2016.

[SID12] M. A. Sehili, D. Istrate, B. Dorizzi, and J. Boudy. "Daily sound recognition using a combination of GMM and SVM for home automation." *European Signal Processing Conference*, pp. 1673–1677, 2012.

[SSB13] Bjrn Schuller, Stefan Steidl, Anton Batliner, Alessandro Vinciarelli, Klaus Scherer, Fabien Ringeval, Mohamed Chetouani, Felix Weninger, Florian Eyben, Erik Marchi, Marcello Mortillaro, Hugues Salamin, Anna Polychroniou, Fabio Valente, and Samuel Kim. "The INTERSPEECH 2013 computational paralinguistics challenge: social signals, conflict, emotion, autism." *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, pp. 148–152, 2013.

[STB12] G Sárosi, B Tarján, A Balog, T Mozsolics, P Mihajlik, and T Fegyó. "On modeling non-word events in large vocabulary continuous speech recognition." *3rd IEEE International Conference on Cognitive Infocommunications (CogInfoCom)*, pp. 649–653, 2012.

[SVS15] Tara Sainath, Oriol Vinyals, Andrew Senior, and Hasim Sak. "Convolutional, Long Short-Term Memory, fully connected Deep Neural Networks." *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pp. 4580–4584, 2015.

[TGP16] Naoya Takahashi, Michael Gygli, Beat Pfister, and Luc Van Gool. "Deep convolutional neural networks and data augmentation for acoustic event detection." 2016.

[TN06] Andrey Temko and Climent Nadeu. "Classification of acoustic events using SVM-based clustering schemes." *Pattern Recognition*, pp. 682–694, 2006.

[UBC12] Burak Uzkent, Buket D. Barkana, and Hakan Cevikalp. "Non-speech environmental sound classification using SVMs with a new set of features." *International Journal of Innovative Computing, Information and Control*, pp. 3511–3524, 2012.

[VA12] Xavier Valero and Francesc Alias. "Gammatone cepstral coefficients: biologically inspired features for non-speech audio classification." *IEEE Transactions on Multimedia*, pp. 1684–1689, 2012.

[Zha16] Peter G Zhang. "Robust audio event recognition with 1-Max pooling convolutional neural networks." *Data Mining and Knowledge Discovery Handbook*, pp. 487–516, 2016.

[ZMS15] Haomin Zhang, Ian McLoughlin, and Yan Song. "Robust sound event recognition using convolutional neural networks." pp. 559–563, 2015.