Speech Emotion Recognition

Weijia Cao wcao34@wisc.edu
ChuYang Chen cchen466@wisc.edu
Xiu Xie xxie65@wisc.edu

Abstract

Speech Emotion Recognition is a task to identify emotions from speech signals. While humans are inherently more capable of telling emotions apart, detecting speech emotions is much more of an arduous task for machines. In the human machine interface application, speech emotion recognition has been a popular research topic for many years. The reason behind such popularity is that this technique has plenty of meaningful real-world applications such as voice-controlled devices. Here, we use deep learning models to build an algorithm that can successfully recognize emotions from human voice signals. We use a dataset from Kaggle which contains over 10,000 voice examples with seven distinct emotions: angry, happy, sad, neutral, fearful, disgusted, surprised. We use Librosa, a Python package for music and audio analysis, to transfer the voice data into Mel-Frequency Cepstral Coefficients (MFCC), which can be put into various deep learning models for further analysis. In this project, we implement several deep learning models and the Convolutional neural network (CNN) turns out to have the best performance with a test accuracy of 62.3%.

1. Introduction

Speech is one of the fastest and most efficient means of human communication. People use speech with one another every day to exchange ideas, express feelings, and achieve a lot more. Although verbal communication is often considered the most crucial part of speech, non-verbal communication, such as speech emotion, also plays an essential role in the efficiency of communication. Correctly recognizing the speech emotion makes communication much more efficient. While humans are inherently more capable of telling emotions apart, detecting speech emotions is much more of an arduous task for machines, which gives human-machine communication a lot of room for improvement. The purpose of this study is to develop a deep learning algorithm that can successfully recognize basic human speech emotions such that human-machine communication can be improved.

In the field of speech emotion recognition, popular methods include fundamental frequency, energy contour, voice quality, and duration of silence. The common features extracted from voice signals are short time log frequency power coefficients (LFPC), linear prediction Cepstral coefficients (LPCC), and Mel-Frequency Cepstral Coefficients (MFCC). In this project, we decided to use MFCC as the feature since MFCC has been the dominant feature used for speech emotion recognition in many previous studies.

It is worth noting that there are several factors contributing to the difficulty of speech emotion recognition: it is still not clear which particular speech features critical to the speech emotion recognition process are. Given high variations in speech styles, speaking rates, speakers, and sentences, the same utterance might represent distinct emotions. Another factor making speech emotion recognition a hard task would be the diversity in culture and societal environment. The same utterance expressed by speakers from different cultures might represent different emotions, and thus it becomes a challenging task when we take changes in the environment and culture into consideration.

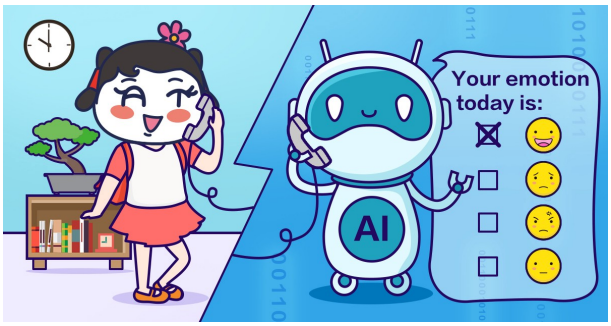

Figure 1. [10] Speech Emotion Recognition enables more efficient human-machine interactions in distance.

1.1. Motivation

The interpretation of emotions is critical among interpersonal relationships and therefore also important in human-machine interfaces. Speech is one major channel for emotion recognition, especially when the other channels, such as facial expression, gesture, and posture, are inaccessible. The ability to recognize emotions from speech also

enables more efficient interpersonal and human-machine interactions in distance. For instance, implementing SER algorithms may help voice-controlled devices better understand the needs of their users without having to ask. Such techniques can also be used in marketing, where emotions expressed in speech can be detected to predict customer attitudes and actions and to help intelligent agents to provide customers with sensible feedback. In addition, adopting SER algorithms in intelligent in-car systems might help detect the driver's mental state and take preventive actions when the driver is not in the best state to drive (e.g., sleepy, angry, etc.) [9].

Figure 2. [6] Speech Emotion Recognition allows in-car systems to take preventive actions when the driver is in a mood dangerous for driving.

2. Related Work

There is an abundance of previous works that renders insights into the topic of this study. In particular, the study conducted by Ingale and Chaudhari [4] demonstrated the feasibility of detecting the emotions and the mental states from non-verbal expressions in speech. They used Mel Frequency Cepstral Coefficients (MFCC) and Linear Predictive Cepstral Coefficients (LPCC) as input features to conduct the study, which is quite similar as our project. However, their study is much more systematic than ours. They split the speech emotion dataset into two parts based on gender, whereas in our project, we are using a comprehensive dataset with a mix of male and female voice messages. What is more, Ingale and Chaudhari's study [4] are divided into two parts: speaker dependent and speaker independent. In this project, the speech files were recorded by 91 actors, and thus it is solely a speaker independent study that we are conducting.

Recently, scientific community has experienced the exponential growth of neural networks due to the advances in hardware and the higher availability of data for training. For instance, Huang et al. [3] have used Convolutional Neural Networks (CNNs) in deep learning systems to conduct a study for Speech Emotion Recognition (SER). Specifically, they utilized semi-CNN to train the dataset, and the performance of their model was outstanding. The recognition is stable and robust even under some unfavorable circumstances (speaker with background noise). Huang's study also shares some similarity with our project because CNN is one of the algorithms being implemented in this study as well.

The experiments and conclusions from previous works as such provide this study with a firm theoretical foundation. More specifically, they validate our choice of applying deep learning techniques to this subject and using MFCC, convolutional neural networks, and recurrent neural networks in our models.

3. Proposed Method

Several deep learning algorithms were implemented to predict the dominant emotion given a snippet of speech data. After extensive data transformation and feature engineering, several state-of-art deep learning algorithms were implemented and their performances were compared in terms of predictive accuracy. First of all, a logistic regression was used as a benchmark model. Then, a Convolutional Neural Networks (CNN) model was implemented to see whether it could achieve a better performance since it works well with image data and we were able to convert speech signals into images through the use of MFCC. In addition, a Recurrent Neural Networks (RNN) was trained to examine whether a more complex classifier could further improve the accuracy. Recurrent neural networks are significantly useful when dealing with temporal sequences of information because they are able to model temporal dependencies introducing a feedback loop between the input and the output of the neural network.

3.1. Multinomial Logistic Regression

We used a multinomial Logistic Regression (also known as Softmax Regression) classification algorithm as a baseline model in this study. In logistic regression we try to predict the probability that a given example belongs to a specific class t versus the probability that it does not; i.e., we want to estimate the probability of the class label taking on each of the h different possible values. In softmax regression, we use the softmax activation function:

$$P(y = t|z_t^{[i]}) = \frac{e^{z_t^{[i]}}}{\sum_{j=1}^{h} e^{z_t^{[i]}}},$$

which is an exponential function that normalizes the activations so that they sum up to 1. Here, we may interpret activations as class-membership probabilities, given that class labels are mutually exclusive.

3.2. Convolutional Neural Networks

A Convolutional Neural Network (CNN) is a deep learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. A convolutional neural network consists of an input layer, hidden layers and an output layer. In any feed-forward neural network, any middle layers are called hidden because their inputs and outputs are masked by the activation function and final convolution.

In addition to convolutional layers, there are also pooling layers, which reduce the dimensions of data by combining the outputs of neuron clusters at one layer into a single neuron in the next layer, and fully-connected layers, which connect every neuron in one layer to every neuron in another layer and work like a multi-layper perceptron.

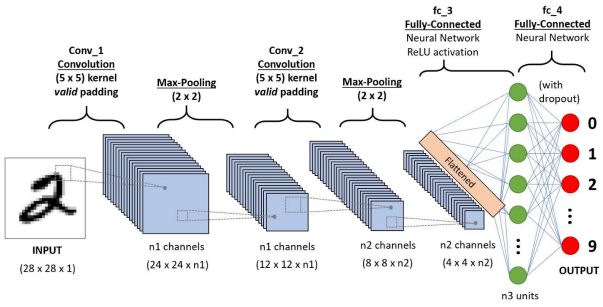

Figure 3. Elements of a Convolutional Neural Network

3.3. Recurrent Neural Networks

Recurrent neural networks (RNN) are a type of neural networks that allow previous outputs to be used as inputs while having hidden states. RNN is widely used in text classification, acoustic modeling, and language translation. Due to the fact that Recurrent Neural Networks are suitable for large datasets and sequential data, we attempted to apply RNN in this project. Specifically, we aim to implement the Long short-term memory (LSTM) algorithm, a deep learning system that avoids the vanishing gradient problem. A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell. LSTM is normally augmented by the recurrent forget gates, and it prevents backpropagated errors from vanishing or exploding [7].

3.4. Transfer Learning

One project that was released online and used a similar dataset to the one we used is the Speech Emotion Recognition with CNN project by Ritzing posted on Kaggle. [8].

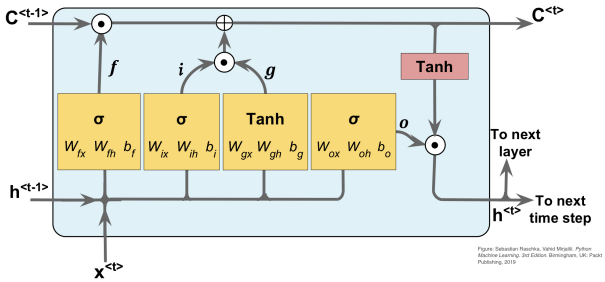

Figure 4. Long-Short Term Memory

The project was incomplete but showed a tendency to have a fair prediction performance. Therefore, we decided to borrow this pre-trained model on our dataset.

Transfer learning is a machine learning method where a model developed for a task is reused as the starting point for a model on a second task. It is a popular approach in deep learning where pre-trained models are used as the starting point on computer vision and natural language processing tasks given the vast compute and time resources required to develop neural network models on these problems and from the huge jumps in skill that they provide on related problems.Lisa Torrey and Jude Shavlik in their chapter on transfer learning describe three possible benefits to look for when using transfer learning [1]:

- Higher start. The initial skill (before refining the model) on the source model is higher than it otherwise would be.
- Higher slope. The rate of improvement of skill during training of the source model is steeper than it otherwise would be.
- Higher asymptote. The converged skill of the trained model is better than it otherwise would be.

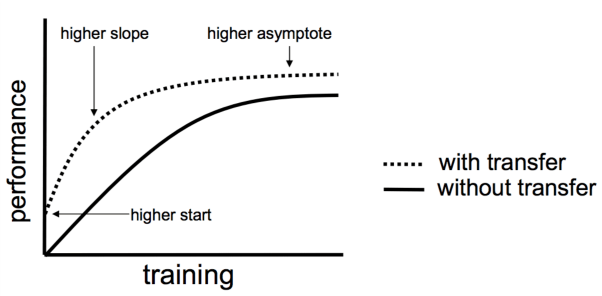

Figure 5. Three Ways in which Transfer Might Improve Learning

4. Experiments

4.1. Dataset

The dataset, which was obtained from Kaggle (`https://www.kaggle.com/uldisvalainis/`

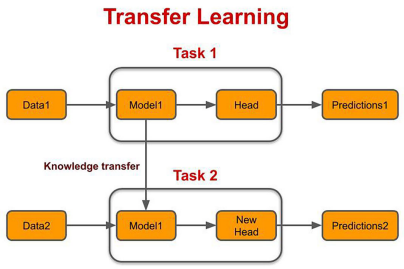

Figure 6. Transfer learning

audio-emotions), contains files from four major data sources – namely, RAVDESS, CREMA-D, SAVEE, and TESS. The dataset consists of original clips from 91 actors. These clips were from 48 male and 43 female actors between the ages of 20 and 74 coming from a variety of races and ethnicities (African America, Asian, Caucasian, Hispanic, and Unspecified). Actors spoke from a selection of 12 sentences. The sentences were presented using one of six different emotions. In total, there are 12,798 examples and 7 labels. Speech recordings are in $.wav$ format and are sorted by emotions:

- Angry - 2167 records (16.7%)
- Happy - 2167 records (16.46%)
- Sad - 2167 records (16.35%)
- Neutral - 1795 records (14.26%)
- Fearful - 2047 records (16.46%)
- Disgusted - 1863 records (15.03%)
- Surprised - 592 records (4.74%)

The dataset was shuffled and split into three sets – 70% went to the training set, 10% went to the validation set, and 20% went to the test set. Normalization was applied to the design matrix, in order to avoid the scenario in which inputs are on very different scales, and thus some weights update more than others, leading to difficulty in convergence.

Feature extraction is the first step in audio data analysis. The correct representation of the different acoustic classes is influenced by the set of acoustic features used in the system, both in frequency and time domains. Consequently, the audio files were converted to numerical data using Mel Frequency Cepstral Coefficient (MFCC). This method frames the audio signal into 20–40ms frames, extracts necessary features from the audio data and performs audio classification.

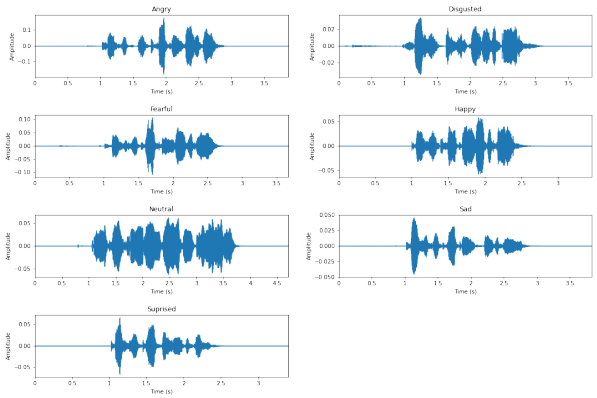

Figure 7. Amplitude plots for every type of the voice emotion in the dataset

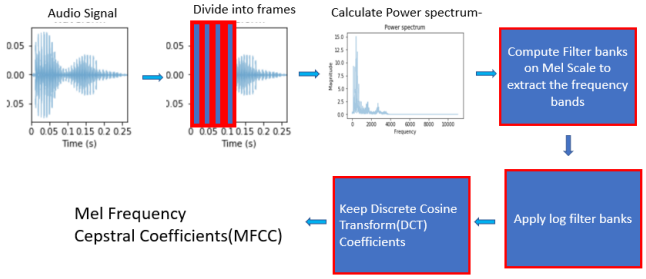

Figure 8. [5] Audio analog to digital signal transformation using MFCC.

4.2. Software

Packages PyTorch, NumPy, Pandas, matplotlib, Scikit-Learn, and Librosa were used for audio and music signal processing. Models used in this study were implemented in Python, through the collaborative notebook-writing platform Google Colaboratory.

4.3. Hardware

The three of us completed this project on our own laptops.

5. Results and Discussion

5.1. Multinomial Logistic Regression

Given that this is a multi-class classification problem, we chose to apply Softmax / Multinomial Logistic Regression. Using the nn.Modual API in PyTorch, with learning rate set to 0.1 and number of epochs set to 600, we were only able to achieve a test accuracy of 0.19, which is slightly below expectation, but reasonable for a baseline model given the number of classes.

We infer that the low accuracy is partly due to Softmax Regression's disregard of the sequentiality in our data. Another significant drawback of Multinomial logistic regression is that it assumes independence among the class labels, whereas in reality there could potentially be a lot of overlap among the 7 emotions studied in this project. For instance, it would make sense for tones of surprise and happiness to sound similar.

5.2. Convolutional Neural Networks

Since 1D CNNs are useful for sequential data, we applied them in this project. Since Dropout can make the network learn not to rely too heavily on particular connections, and can thus make the model more robust and make the weight values more spread out, 2 Dropout layers were included in this model. In addition, given that Batch Normalization might help reduce internal covariate shift or provide more parameters that would help layers learn a little bit more independently, we employed 2 layers of Batch Normalization in our model. We will show later in this report that both optimization algorithms turned out to improve the final performance of our model.

Batch Normalization

Step 1:

$$\mu_j = \frac{1}{n}\sum_i z_j^{[i]}$$

$$\sigma_j^2 = \frac{1}{n}\sum_i (z_j^{[i]} - \mu_j)^2$$

$$z_j'^{[i]} = \frac{z_j^{[i]} - \mu_j}{\sigma_j + \epsilon}$$

Step 2:

$$a_j'^{[i]} = \gamma_j z_j'^{[i]} + \beta_j$$

BatchNorm1d (and BatchNorm2d):

$$y = \frac{x - E[x]}{\sqrt{Var[x] + \epsilon}} * \gamma + \beta$$

5.3. Recurrent Neural Networks

Specifically, we tried to build our model based on an image classification project using RNN from Kaggle. [2] Unfortunately, we encountered some problems with the code on the training stage that we were not able to address. The input batch size was not able to match the target batch size. Further research needs to be done to figure out this issue.

5.4. Transfer Learning

Applying transfer learning, we used a previously-trained CNN model on our dataset. The final performance was demonstrated with an accuracy of 52.99%, which is lower than that of our own CNN model.

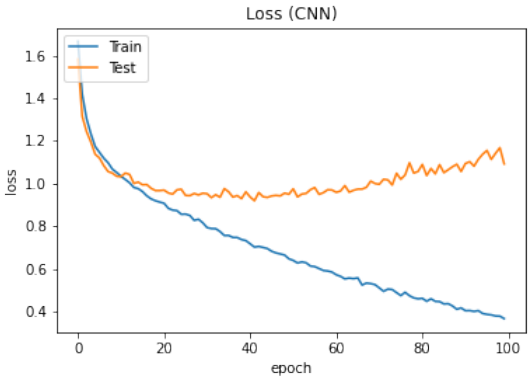

Figure 9. Loss for CNN

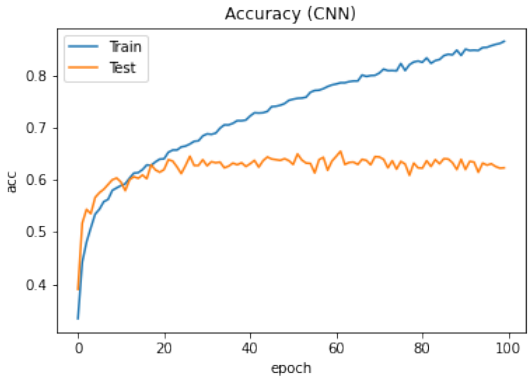

Figure 10. Accuracy for CNN

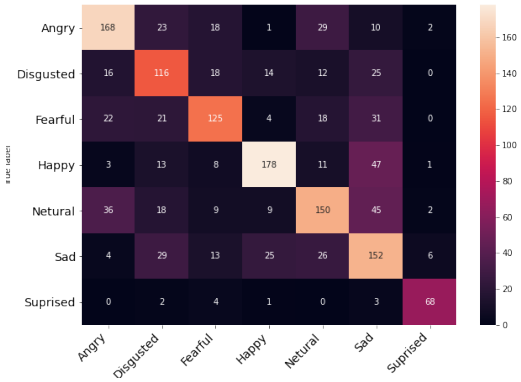

Figure 11. Confusion Matrix for CNN

The accuracy demonstrated in the original project on Kaggle was 50.21%. We were surprised to find that the test accuracy was actually higher when this model was adapted to our dataset. Perhaps this is due to the fact that our data is more organized and normalized.

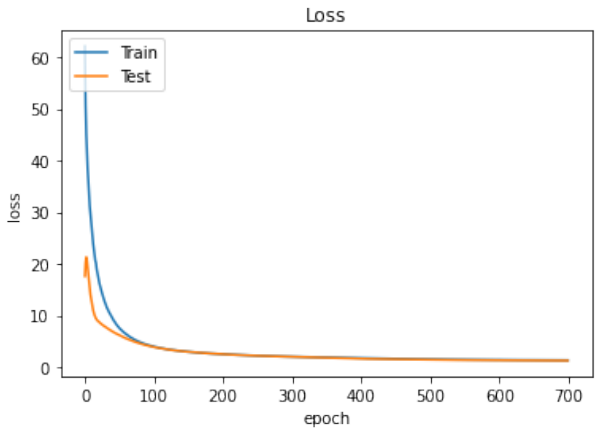

Figure 12. Loss for transfer learning (CNN)

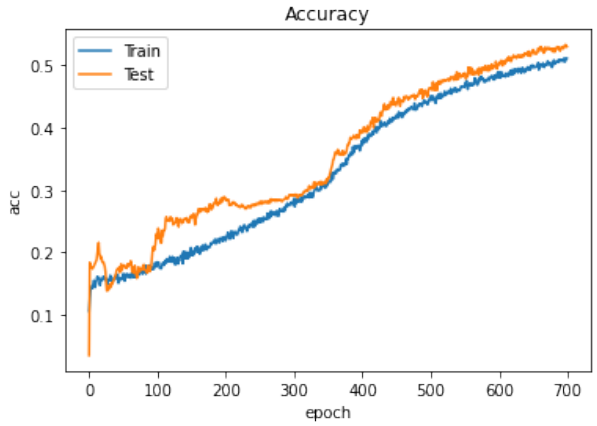

Figure 13. Accuracy for transfer learning (CNN)

Method	Accuracy
Logistic	19.0 %
CNN	62.3 %
Transfer	53.0 %
RNN	N/A

Table 1. Comparison of different models.

6. Conclusions

As we expected, speech emotion recognition is still a difficult task for machine. The highest test accuracy we achieve is from the Convolutional neural network (CNN). It yields a relatively high training accuracy (86.79%) and a lower test accuracy (62.30%). We've spent plenty of time on hyperparameter tuning but the test accuracy is always around sixty percent. It's quite obvious that we are facing a severe overfitting problem. To solve this problem, we include dropout method in our CNN model, and we also decrease the learning rate so that it might be able to pass the local maximum. However, those methods didn't work. The problem might be the size of data. We only have around 10 thousand pieces of very short voice messages, which is not enough for deep learning model. Thus, we are still looking for more dataset to improve the model performance in the future.

7. Acknowledgements

Special thanks to Kaggle user Uldis Valainis for compiling the datasets together and assigning emotion memberships. We would also like to pay tribute to the owners of the four original datasets – RAVDESS, CREMA-D, SAVEE, and TESS – for providing the valuable resources, and Professor Sebastian Raschka for the support he has provided us throughout the course of this project. The Kaggle community and the PyTorch documentation has also proven to be extremely helpful when implementing the different models.

8. Contributions

The team distributed the workload evenly throughout the entire project, from the initial proposal write up to the final report. Communication was easy between us and each of us adhered to the timelines and deadlines that were agreed upon during our group meetings on Zoom. As for the actual project, all of us worked on one model each. Specifically, Xie worked on the baseline Softmax Regression model, Chen worked on the Convolutional Neural Networks model, and Cao worked on the Recurrent Neural Networks model.

For the project write-up, each of us was in charge of reporting the model worked on. Furthermore, Chen summarized our study into the abstract paragraph and wrote on related works in the field. Xie wrote the parts of the introduction, the dataset, and the acknowledgement of this report. Cao worked on the proposed methodology, the results and discussion, and the contribution.

During the presentation, we decided to work collaboratively on Google Slides and divide roles to play in terms of logistics and flow control. As for the report, we each wrote about the models we worked on and split up the remainder of what was left among us. Overall, the workload was spread evenly and it was a pleasant group work experience.

References

[1] J. Brownlee. A gentle introduction to transfer learning for deep learning. *Deep Learning for Computer Vision*, 2019.

[2] DATAI. Recurrent neural network with pytorch, Apr 2020.

[3] Z. Huang, M. Dong, Q. Mao, and Y. Zhan. Speech emotion recognition using cnn. pages 801–804, 2014.

[4] A. B. Ingale and D. Chaudhari. Speech emotion recognition. *International Journal of Soft Computing and Engineering (IJSCE)*, 2(1):235–238, 2012.

[5] R. Khandelwal. Deep learning audio classification, Dec 2020.

[6] A. Ramkumar. Look out: There are 160 million angry drivers on the road, Jul 2016.

[7] S. Raschka. Lstm cell, Apr 2021.

[8] Ritzing. Speech emotion recognition with cnn, Nov 2019.

[9] Shivamburnwal. Speech emotion recognition. *Kaggle*, May 2020.

[10] A. Tech. An ensemble framework of voice-based emotion recognition, Apr 2018.