Distress Detection from Speech Data

Amit Kumar Yadav, Saurabh Swaroop
Indiana University
Bloomington, Indiana

ABSTRACT

In real life, there is plenty of unstructured data available which largely remains unexploited. For example, while we communicate with people, we not only give attention to the words but to the emotions, body language, context, etc. At times processing these unstructured data can reap us easy rewards. While there is extensive research being conducted in areas such as speech and music, work on the emotion detection like Distress in human sound is scarce. Distress is defined as feeling of anxiety, fear and pain. Whenever a human being is in distress, it is reflected in their voice. Distress detection has many real life applications for elderly, people under protection, people living in dangerous areas, people travelling alone at night, etc. In this project, we are trying to detect distress from speech data using several artificial neural network network architectures.The aim of this paper is to identify and extract features to create efficient deep learning models for Distress Detection from Speech Data in real-life conditions.

KEYWORDS

Distress Detection, CNN, MFCC, Pitch, Zero Crossing Rate, Root Mean Square Energy, Ensemble

1 INTRODUCTION

In recent years, automatic classification of emotions out of sound files is a growing research field, and it has become an important factor for various emerging applications, and therefore it has gained large focus. In this project, we have decided to perform Distress Detection from speech signals. We were inspired to this project from work of Alkaher et. al. in 'Detection of Distress in Speech'[1]. Similar to their work, we treat the task of Distress Detection as binary classification problem in which we are trying to separate speech signals carrying emotions of *anger* and *fear* as distress whereas every other emotion as non-distress signals. We are using neural network models instead of a Support Vector Machine in this work. We examine various sets of audio features in their effectiveness for the given task. Furthermore, we are performing a secondary binary classification task in which we are trying to separate audio signals belonging to anger and fear class.

2 FEATURE EXTRACTION

The extraction of the best parametric representation of sound signals is an important task to produce a better recognition performance. The efficiency of feature extraction affects the accuracy of the model and so it is very important to choose the right feature for good performance of the model.We examine the efficacy of following set of features for the two task defined above:

- Raw Time Domain Signal
- Spectral Features (MFCC and STFT)
- Continuous Features (Pitch, Zero Crossing Rate and Root Mean Square Energy)

Our observations of the individual performance of these features in identifying temporal patterns are listed in a separate conclusion section.

2.1 Raw Time Domain Signal

The raw audio signal is a time-domain representation of continuous source signal. We choose to perform distress detection using raw audio signals to determine the performance we could achieve without doing any feature extraction.

2.2 Spectral Features

2.2.1 MFCC. - Mel Frequency Cepstral Coefficients (MFCCs) is most common feature used for automatic sound recognition. They were introduced by Davis and Mermelstein in the 1980's, and have been state-of-the-art ever since. MFCC coefficients are obtained by taking a discrete cosine transform of a log power spectrum on a nonlinear mel scale of frequency [8]. MFCC features have been successful in audio processing tasks because they closely capture the human-auditory response system. We use librosa library to extract MFCC features from raw audio data. We take 20 MFCC coefficient along with their first (delta) and second derivatives (delta-delta) for each of the audio file.

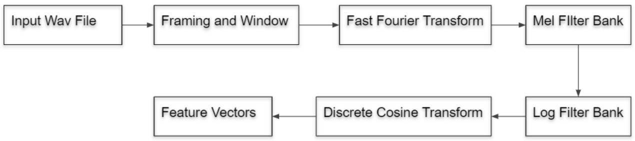


Figure 1: MFCC Steps

2.2.2 STFT. - The short-time Fourier transform (STFT), is a Fourier-related transform used to determine the sinusoidal frequency and phase content of local sections of a signal as it changes over time.In practice, the procedure for computing STFTs is to divide a longer time signal into shorter segments of equal length and then compute the Fourier transform separately on each shorter segment. This reveals the Fourier spectrum on each shorter segment [3].

2.3 Continuous Features

2.3.1 Pitch. - Pitch is a perceptual property that allows the ordering of sounds on a frequency-related scale. Pitch may be quantified as a frequency, which is referred as Fundamental frequency (F0). Pitch and pitch changes in words form the tone of a tonal language, such as Chinese.

2.3.2 Zero-crossing rate. - It is a measure of number of times in a given time interval/frame that the amplitude of the speech signals passes through a value of zero.

2.3.3 Root Mean Square Energy. The root-mean-square Energy is used to capture the energy carried by the sound wave, which is called the intensity. The intensity of a sound wave is the average amount of energy transmitted per unit time through a unit area in a specified direction. In other words it corresponds to the root mean square of total magnitude of the signal for each time frame. For audio signals, that roughly corresponds to how loud the signal is. [4]

3 NETWORK ARCHITECTURE

Convolution Neural Network - In deep learning, a convolution neural network (CNN, or ConvNet) is a class of deep neural networks, most commonly applied to analyzing visual imagery. CNN are regularized versions of multilayer perceptrons. We chose to use CNN for our case because they have proven to work well for time series/sequential data and CNN models are easier and faster to train than a RNN model.

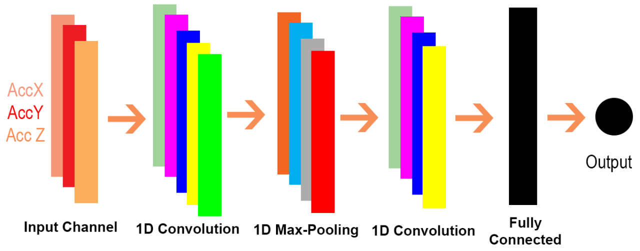


Figure 2: An example CNN - Architecture [5]

Recurrent Neural Network - A recurrent neural network (RNN) is a class of artificial neural network where connections between nodes form a directed graph along a temporal sequence. This allows it to exhibit temporal dynamic behavior.[6] Unlike feed forward neural networks, RNNs can use their internal state (memory) to process sequences of inputs. This makes them applicable to tasks such as unsegmented, connected handwriting recognition or speech recognition. RNN architecture is meant to to utilize sequential information. In a traditional neural network we assume that all inputs (and outputs) are independent of each other.RNNs are called recurrent because they perform the same task for every element of a sequence, with the output being depended on the previous computations.[7]

Our Approach - We tried various RNN and CNN model architectures and found that, for both the tasks, for every feature set, CNN models performed equally good as RNN models or better. CNN models were faster to train as well. All these reasons made us choose CNN model over RNN models for most of the feature set in both tasks. We used pooling to reduce the dimensionality of data and dropout to avoid over-fitting. We also tried combination of CNN and RNN layers and have used it in one of models.

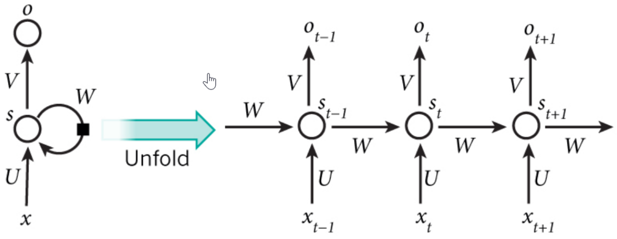


Figure 3: Example of a recurrent neural network with its forward computation

[7]

4 DATASET

CREMA-D[2] is a data set of 7,442 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 (Anger, Disgust, Fear, Happy, Neutral, and Sad) and four different emotion levels (Low, Medium, High, and Unspecified).

Participants rated the emotion and emotion levels based on the combined audiovisual presentation, the video alone, and the audio alone. Due to the large number of ratings needed, this effort was crowd-sourced and a total of 2443 participants each rated 90 unique clips, 30 audio, 30 visual, and 30 audio-visual.

We segregated the data set in two parts for two classification experiments. In first part, we divided the audio files in two groups of Distress and Non-Distress classes. This experiment was to observe the model performance and feature importance to classify Distress vs non-distress emotion in an audio file. In second part, we took all the audio files belonging to distress class and divided them into two classes of distress Fear and Anger. This experiment was to observe how model performed in identifying between different distress signals.

4.1 Pre-Processing

We did pre-processing in following steps.

Analysis on file duration :When we analyzed the files, we found out that length of the files range from 2 Sec to 4 sec. As majority of the sound files duration range in 3 - 4 secs, we decided to convert all audio files to 3 sec long. We do this by padding shorter clips or by clipping files with longer duration.

Hyper-parameters : We tried with higher sampling rates but due to performance issues, we chose SR to be 8000. For Fourier transform, hop length taken was 512 and number of MFCC frames was 20. Learning rate of 0.001 and He initialization was used for training CNN.

File format : Numpy .npy file format was used for faster reading and loading of data during multiple training experiments.

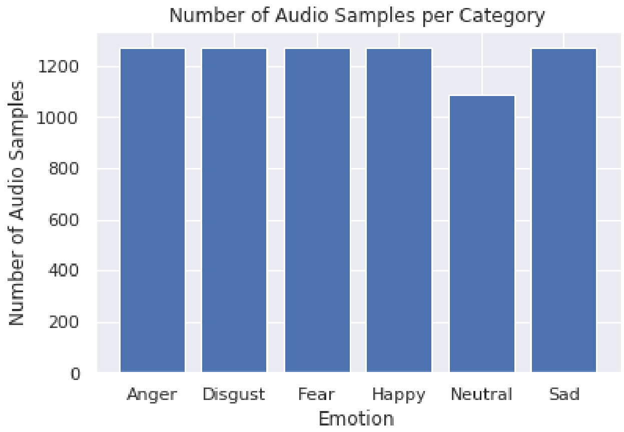


Figure 4: Class Distribution

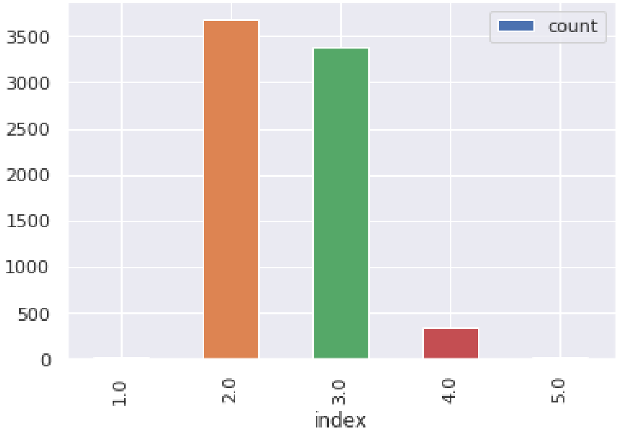


Figure 5: Time duration of audio files

Training splits : Training and Evaluation splits were done to calculate accuracy on new data.

5 EXPERIMENTS

5.1 Distress vs Non-Distress

The first classification experiment we performed was to classify Distress vs Non-Distress classes. We segregated the data set in two classes. First class had all the audio files belonging to Distress (Fear and Anger). Other class had all audio files from Non-Distress (Happy, Neutral, Sad, Disgust). Since the samples belonging to *OTHER* class were more than that in distress class, we used down-sampling to make the dataset balanced. We trained all the samples with four different feature sets and measured the performance.

5.2 Fear vs Anger

The second classification experiment we performed was to classify Fear vs Anger classes. We segregated the data set from Distress class in two sub-classes. First class had all the audio files belonging to Fear. Other class had all audio files belonging to Anger. We trained all the samples with four different feature sets and measured the performance.

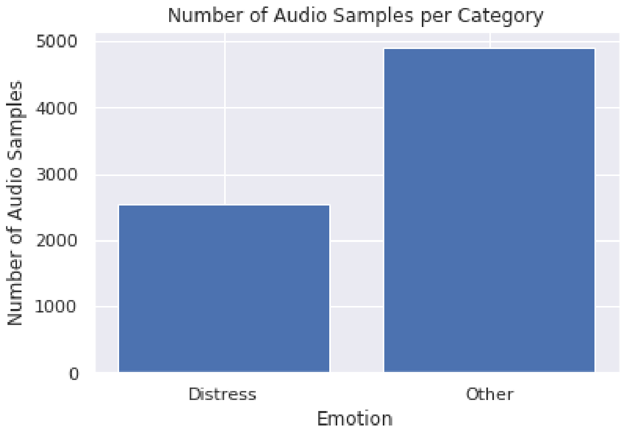


Figure 6: Distress-Non Distress data

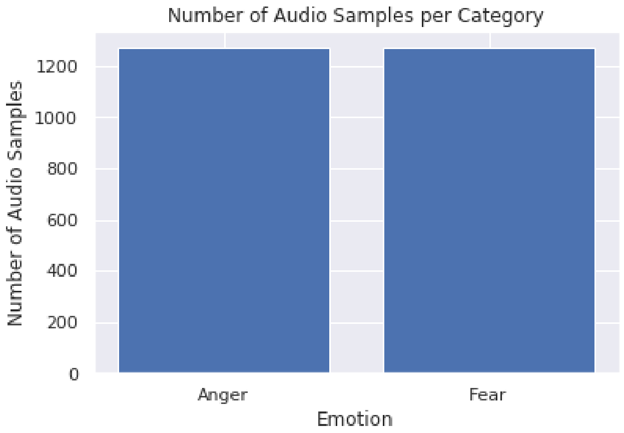


Figure 7: Anger-Fear data

5.3 Feature sets

1. Raw audio time steps with extended dimension
2. STFT
3. MFCC, MFCC DElta and MFCC Delta Delta stacked together
4. RMSE, STFT and Pitch stacked together

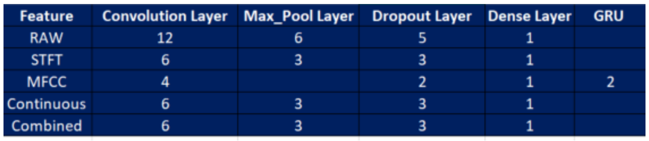
5.4 Ensemble Learning

Ensembling is a powerful technique which is used to create a strong model from multiple weak models. We have decided to use a weighted average ensemble such that the contribution of each weak model is weighted by how accurate it is on held-out test data. Ensemble didn't help us improve accuracy in our case.

5.5 Combined Feature

After training and testing the model with each individual feature set, we stacked together all features and created a combined feature set to train and test the data. We observed that the best accuracy was achieved with this technique for both the cases.

6 MODELS



Feature	Convolution Layer	Max_Pool Layer	Dropout Layer	Dense Layer	GRU
RAW	12	6	5	1	
STFT	6	3	3	1	
MFCC	4		2	1	2
Continuous	6	3	3	1	
Combined	6	3	3	1	

Figure 8: Model configurations for different feature sets

7 RESULT AND ANALYSIS

Our best prediction on training the two classification models with four different feature sets are as below:

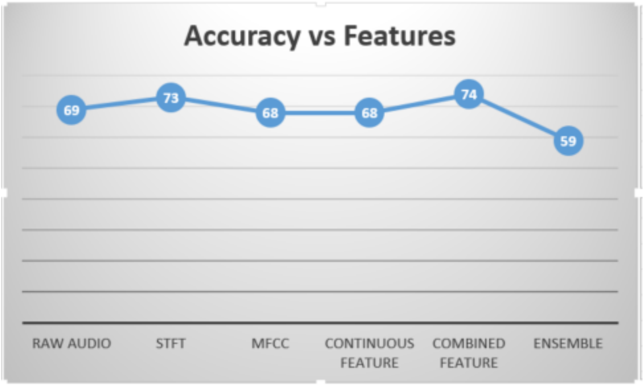


Figure 9: Accuracy on different feature sets for Distres vs Non Distress

There are many factors which could be more experimented with to achieve better accuracy:

Similar Temporal structure - Our CNN is trying to identify difference in temporal structures of the spectrogram of different voice emotions. Poor accuracy in the case of Distress vs Non-Distress could be due to temporal similarity between multiple classes. That could be the reason the network got confused and needs to be researched further. In the case of two distress type class classification, accuracy achieved is good and the network has been successfully able to identify the temporal pattern difference between the two classes.

Feature Extraction - We created four different feature sets to analyze the contribution of each one in network ability to classify correctly. But for almost most of the cases we got similar results from all the feature sets. We think, more feature sets need to be explored to improve the performance.

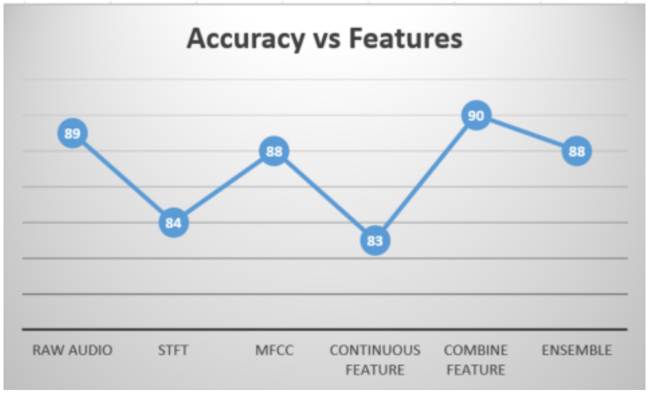


Figure 10: Accuracy on different feature sets for Fear vs Anger

8 FUTURE WORK

We look forward to dive more into distress classification related research as follow.

- We would like to experiment with different dataset and see if the low accuracy that we got for distress classification is due to limitations of the dataset or the models that we are using. Furthermore, all audio signals in CREMA-D dataset are captured in a very controlled environment and lack the problems that come with a recording in real environment. We would like to experiment with a dataset that closely captures real-distress scenario.
- Experiment more with the model architectures.
- Experiment with more speech features.

9 CONCLUSION

In this project, we have implemented various models that take a specific set of audio features as input and perform binary classification on input data. We are interested in two types of classification tasks: Distress Vs. Non-Distress and Anger Vs. Fear. The first classification task is our main objective and covers a broad range of emotions in the two classes. We tried various network architectures but couldn't achieve a very high accuracy for this task. The second classification task is our secondary objective in which we are trying to separate the distresser from the distressed individual. We were able to achieve very good results for the second task using very simple neural network models. We also tried to compare how an ensemble of various models, working on specific set of features, would fare against a model that utilizes all the audio of these features combined into a single vector. We conclude that the model that uses all audio features was more accurate than the ensemble in both the tasks.

ACKNOWLEDGMENTS

We want to thank Professor Minje Kim for his continued support, guidance, and encouragement throughout the semester. We sincerely appreciate the learning opportunities provided by Professor Minje Kim and his teaching assistants who constantly motivated us to go beyond our limits. This project would not have been complete without their dedicated support and our sincere thanks to all our fellow students.

REFERENCES

[1] Yehav Alkaher, Osher Dahan, and Yair Moshe. 2016. Detection of distress in speech. In *2016 IEEE International Conference on the Science of Electrical Engineering (ICSEE)*. IEEE, 1–5.

[2] Houwei Cao, David G Cooper, Michael K Keutmann, Ruben C Gur, Ani Nenkova, and Ragini Verma. 2014. CREMA-D: Crowd-sourced emotional multimodal actors dataset. *IEEE transactions on affective computing* 5, 4 (2014), 377–390.

[3] Wikipedia. [n. d.]. Short-time Fourier transform. ([n. d.]). "https://en.wikipedia.org/wiki/Short-time_Fourier_transform"

[4] Wikipedia. [n. d.]. Short-time Fourier transform. ([n. d.]). "https://dosits.org/science/advanced-topics/introduction-to-signal-levels/"

[5] Wikipedia. [n. d.]. Short-time Fourier transform. ([n. d.]). "https://stackoverflow.com/questions/48859378/how-to-give-the-1d-input-to-convolutional-neural-networkcnn-using-keras"

[6] Wikipedia. [n. d.]. Short-time Fourier transform. ([n. d.]). "https://en.wikipedia.org/wiki/Recurrent_neural_network"

[7] Wikipedia. [n. d.]. Short-time Fourier transform. ([n. d.]). "http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnns/"

[8] Wikipedia contributors. 2019. Mel-frequency cepstrum — Wikipedia, The Free Encyclopedia. (2019). https://en.wikipedia.org/w/index.php?title=Mel-frequency_cepstrum&oldid=886751555 [Online; accessed 2-May-2019].