CS 484-DL1

Spring ‘21

**Speech Emotion Recognizer (SER)**

By Ahrisa Hwang and Nannan Liu

**Introduction**:

Emotional information in speech is an important behavioral signal that reflects human emotions. Identifying the emotional information contained in speech is an important part of communication between humans and computers. Being able to identify the emotions in speech is also helpful for various professions such as acting, police work, and general conversation analytics. Therefore, the motivation here is to create a classifier that can recognize the emotion in someone’s speech by analyzing the .wav patterns.

**Related Work**:

Neural network models are agreed upon as one of the more effective methods to analyze and classify sound wave data (.wav files) because the features extracted cannot always be compared linearly. With neural network models, more data to compare also raises the effectiveness of the model. The following works have done research or worked with speech related data in a manner similar to our project.

The website, [the python code](https://www.thepythoncode.com/article/building-a-speech-emotion-recognizer-using-sklearn) [1], relies on feature extraction from librosa and then the sklearn neural network, MLPClassifier, to pipeline a SER. First the SER would take the data using soundfile before extracting features from it. The features extracted include mfcc (mel-frequency cepstral coefficients), chroma, mel (MEL Spectrogram Frequency), contrast, and tonnetz. By putting these features into a function there is a control in place to select which features to extract and which to ignore. The train data is partitioned to train and test data for accuracy prediction using sklearn train test split. Then the data is fed into the MLPClassifier with hyper parameters using grid search. Finally, the result is found using accuracy score, the output obtained in this example being 75%. I believe one of the useful things to take from this case is the use of placing features in a function, so the chosen features can be easily adjusted with a function call.

While this is not building a SER, the way the speech classifier is built and how data is extracted from .wav files is worth analyzing. [Toward data science](https://towardsdatascience.com/speech-classification-using-neural-networks-the-basics-e5b08d6928b7) [2] uses librosa.load to get sound wave data instead of soundfile. They note that the sampling rate is an important value to take into consideration. The first accuracy test they do is process the data in a simple MLP neural network with a single hidden layer. Since the test validation accuracy was bad, they switched to making images of the sound wave data (using spectrograms) and used convolutional neural networks for image classification instead. The accuracy goes up to around 65% with the switch, making the spectrogram image classification over the MLP a viable argument to make.

People from RMIT University, Melbourne, VIC, Australia, take a different approach to SER as described in their article, “[Real-Time Speech Emotion Recognition Using a Pre-trained Image Classification Network: Effects of Bandwidth Reduction and Companding](https://www.frontiersin.org/articles/10.3389/fcomp.2020.00014/full#h3)” [3]. Deciding to scrap the feature selection, in this research article they took the route of deep neural network classifiers (DNN). But because DNN requires a lot of data to be effective, the use of pre-trained neural networks is explored. By using pre-trained neural networks, all that needs to be done is to select appropriate network parameters and fine-tune the network using the dataset for your SER. Another important thing to note is that pre-trained neural networks typically use image classification, so a conversion to image data from speech is needed. Preprocessing includes sampling frequency, speech companding, and buffering speech waveforms into blocks. After preprocessing, the next step is generating the spectrogram arrays to convert to RGB images. The pre-trained neural network used is AlexNet, a convolutional neural network pre-trained with over 1.2 million images. Now with both the pre-processed image data of the speech dataset and the pre-trained CNN, it is a matter of fine-tuning AlexNet before experimenting can begin. The results showed about a 3% reduction each in accuracy with sampling frequency reduction and companding with an overall 79.7% average weighted accuracy.

Babak Basharirad and Mohammadreza Moradhaseli compared different SER methods and their results in “Speech Emotion Recognition Methods: A Literature Review” [4]. Although this article is 4 years old, the comparisons provide a good idea on what kind of classification method might be good to apply. The summarized results show the highest recognition rate to be HMM (hidden Markov models) at best 96% and at average 78% while SVM (support vector machine) & RBF (radial basis function) have the best consistent average at 91.6%.

**Solution:**

The steps for our technical approach include loading data, extracting features, setting up the classifier, and running the data into the classifier.

* *Setting up the Dataframe*:

Take the data using os.listdir and loop through each record to take note of their label before setting them in a dataframe. We chose to implement pandas dataframes as it was a straightforward way of organizing and manipulating large datasets.

* *Feature Extraction*:

Using librosa to retrieve the soundwave data and then extracting any or all of the following features: STFT, spectral contrast, chroma, MFCC, root mean square value, or mel. The selected features will be concatted together to create the dataframe that will be fed into the classifier. Using sklearn.model\_selection.train\_test\_split, the data is then split into train and test data. The method call for feature extraction is heavily influenced by a related work [1] as it allowed for easy manipulation of which features to include in our training dataframe.

* *Setting Up Classifier*:

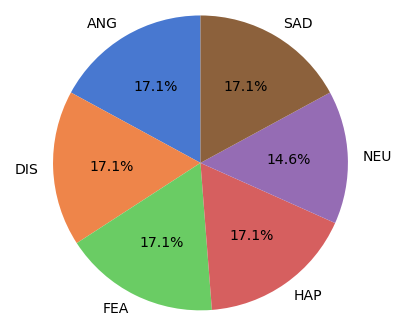
The model we chose to use is the sequential model from the keras library for reasons including the wide acceptance and usage of keras, the tensorflow support that allows processing power to be shared with the GPU, and the custom layers that allow for easier customization of the model. For our model, the layers added include conv1d, activation, dropout, maxpooling1d, flatten, dense, and batchnormalization. Keras is then compiled for accuracy as a metric because the goal for this project is testing for accuracy rather than labeling test data.

* *Run the Data***:**

The model then fits the test data, partitioned earlier, and then evaluates the accuracy of the labeled test data for a quick accuracy output.

**Experiments**:

* **Data**: Crowd Sourced Emotional Multimodal Actors Dataset ([CREMA-D](https://www.kaggle.com/ejlok1/cremad)) [5]
  + # of Records: 7442 records
  + File Size: 577.83 MB
  + Label Percentage:



* **Experimental setup**:
  + Experiment 1: The aim of this experiment is to see what combination of features have the highest impact on the accuracy score. Seeing which features impact the score allow for a more precise choosing on which feature and its metrics to fine tune.
  + Experiment 2: The goal is to see if simpler is better. Reduce the number of layers added onto the keras model and see if that significantly changes the accuracy outcome and in what way.
* **Experimental results and analysis**:

*Experiment 1* - Choosing Feature Extractions

With the number of features being extracted from the sound wave data, oversampling was a concern and runtime as well since unnecessary data could be unintentionally added. But after a series of tests with different feature extractions, the compilation of all the features we implemented provided the best accuracy. Most of the other tests sported an average of 4% accuracy loss except for the run missing the mfcc + mel features. The drop in accuracy is evidence that mfcc and mel have a large influence on speech recognition.

* All features accuracy results:



* Chroma, mfcc, mel:

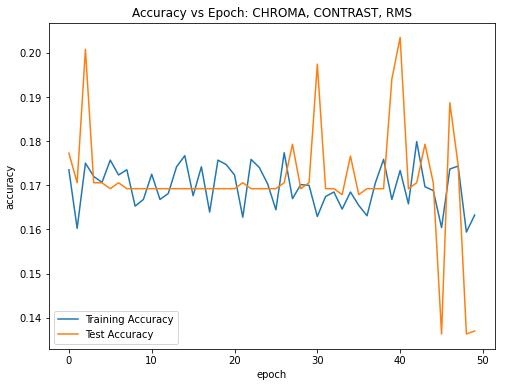


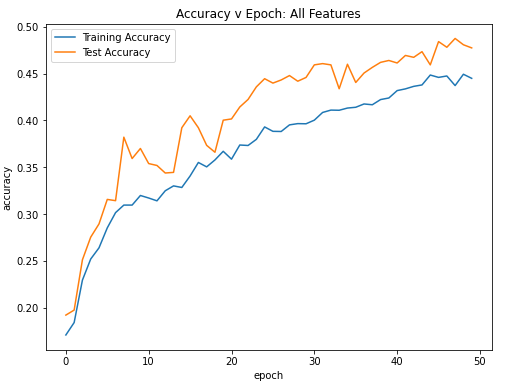
* Contrast, chroma, mel:



* Chroma, contrast, rms:





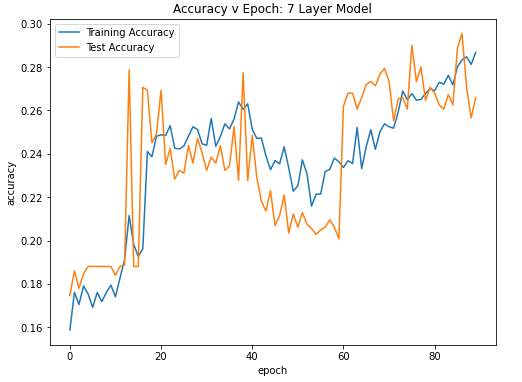


*Experiment 2*: Is simpler better?

There are two models that we tested, one with 17 layers of complexity and another with only 7 layers of complexity. The accuracy dropped sharply with the lower accuracy, demonstrating that simpler may not necessarily be better.

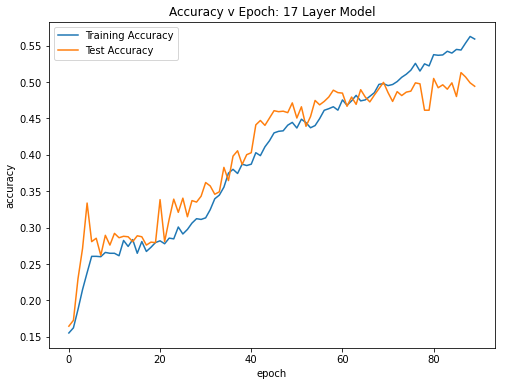
* 7 Layers:





* 17 Layers:





**Challenges and lessons learned & conclusion**:

The emotional information embedded in our speech is a very important behavior signal for human beings when it comes to communication. Accurate identifying that emotional information then becomes important in the effort to improve communication between humans and computers. But considering how it can be difficult for even other humans to pinpoint emotion in speech, it’s not surprising that it is even more complicated for computers to learn to identify emotion. Many nuances go into a person’s voice and they need to be taken into consideration (or eliminated) before a computer can learn the relationship between similar emotional statements and make proper judgements on future speech tests. The difficulty therefore lies in figuring out how to read the data of a sound wave and pinpointing what needs to be filtered to get useful information in classifying emotion in a person’s speech.

**Contribution**:

* Both:
  + Background research needed to build up the system.
  + Decisions on overall schematics of the SER process.
  + Equally worked on presentation slides and recording.
* Nannan Liu
  + Primary coder; laid out the foundation of the code.
  + Report editor: looked over the report and edited/added information as needed.
  + Conducted experiment 2
* Ahrisa Hwang
  + Secondary coder: reorganized the code by putting repeatable code in the speech\_emotion.py file and added some experimentation to the code.
  + Primary on report: structured and wrote out most of the information on the report.
  + Conducted experiment 1

Works Referenced

[1] <https://www.thepythoncode.com/article/building-a-speech-emotion-recognizer-using-sklearn>

[2] <https://towardsdatascience.com/speech-classification-using-neural-networks-the-basics-e5b08d6928b7>

[3] <https://www.frontiersin.org/articles/10.3389/fcomp.2020.00014/full#h3>

[4] <https://aip.scitation.org/doi/pdf/10.1063/1.5005438>

[5] <https://www.kaggle.com/ejlok1/cremad>