The Edward S. Rogers Sr. Department of

Electrical and Computer Engineering

University of Toronto

ECE496Y Design Project Course

Group Final Report

Project title: Automating Speech Emotion Recognition with Machine Learning

Team Number: 2018833

Team 2018833: Xin Geng, kerryn.geng@mail.utoronto.ca

Kejia Huang, katherine.huang@mail.utoronto.ca

Jiaxin Liu, jiaxinjessica.liu@mail.utoronto.ca

Lichuan Zhang lichuan.zhang@mail.utoronto.ca

Supervisor: Jonathan Rose

Section Number: 04

Administrator: Phil Anderson

Submission Date: March 21st, 2019

Executive Summary

Humans express their thoughts and emotions naturally through speech. Enabling natural interactions between humans and machines over a wide spectrum of activities from clinical monitoring to responsive entertainment systems requires automated systems capable of recognizing context and emotion in naturally occurring human speech. Thus, the next-generation human-computer interfaces in these application areas will be empowered by speech-based emotional intelligence.

Classification of emotions can be achieved by analyzing the acoustical content of speech signals. Traditional methods rely heavily on features tuned to acoustical characteristics and produce robust results when combined with supervised learning algorithms. More recent works, on the other hand, are inspired by the learning capabilities of deep-learning algorithms to provide a more holistic solution.

The gap in the current state-of-the-art is a hybrid technique that takes advantages of multiple learning algorithms. In this project, the team leverages existing machine learning methodologies to develop a hybrid emotion classification technique that operates on the acoustical modality of speech.

The system takes in arbitrary input speech signals in standard North American English and identifies the six archetypal emotions: happy, sad, angry, fearful, disgust, and surprised. Based on the assessment of the proposed design and feasibility evaluation, the aim is to achieve a true positive rate of 60%.

The required 60% accuracy is met through a hybrid classifier combining the convolutional and recurrent neural networks (CNN and RNN). With the development of an Android application and a cloud server, an end-to-end system is delivered and performs all required functionalities. Future work involves a showcase activity with a displaying poster at the design fair in early April.

Acknowledgements

We would like to express our special thanks to our supervisor Professor Rose, who gave us the precious opportunity to work with him. Not only he did he guide us with great patience on the project, but also gave us a lot of practical advice on how to prepare and behave as a professional engineer.

Secondly, we would like to thank our administrator Professor Anderson, who guided us on the management side of engineering projects.

We truly appreciate the opportunity to work with the two professional, experienced, and renowned professors. Thank you for their responsibility, patience, and careful guidance. We are lucky to accomplish this project as the closing project in our undergraduate studies, and we will not forget what we learn from it.

Additionally, we thank the staff at the Department of Electrical and Computer Engineering for the great organization and support for the ECE496 course.

Lastly, we would like to say thanks to each of our team members. We have been cohesive and collaborative. We believe that this journey will be a beautiful memory for all of us.

Group Highlights and Individual Contributions

Speech emotion recognition has the potential to revolutionize human-computer interaction incumbent on speech understanding. The team developed models to identify emotions from the acoustic modality of speech. The team also showcased the performance of developed models via an Android application. With all the milestones and deliverables to this date, the project is complete. The project is divided into the following major parts: (1) research and development in speech to emotion algorithm, and (2) development in Android front-end application and a cloud server. The prototype has achieved proposed functionalities and requirements. The team has also met internal deadlines for all individual milestones.

For the speech-to-emotion recognition algorithm research part, the team investigated and experimented on five neural networks. Two of them, the CNN and the CNN+LSTM models achieved an accuracy of 55% on the test dataset. Another final decision layer was added to integrate the two models, produced a final weighted classification result and achieved 60% accuracy on the test data, meeting the project requirement. For the text-based model and audio-based SVM model mentioned in the previous work plan, the test results did not achieve the objective accuracy, therefore these models were not integrated to the final decision neural network.

To test the developed models with real-time recorded speeches, an aesthetically pleasing and user-friendly front-end GUI was created. The application allowed the users to record audio and displayed the analysis result and the transcribed text. The buttons in the GUI are self-explanatory. Other small features such as notification and popup windows were implemented to minimize the learning curve. On the back-end of the GUI, the recorded audio in ".wav" format is transcribed using Watson Speech to Text services and communicate with a server to receive the analysis result. The final choice of the cloud server is AWS EC2 running the Flask microframework, which runs the emotion analysis algorithm and sends back the result as a JSON object.

Kejia Huang

Kejia has contributed to multiple areas in this project. Her active leadership role guided the team to complete the project goal and meet the requirements.

On the research side, she delivered a CNN-based model operating on the spectral representation of audio signals. Applying transfer learning, the accuracy of this model matched state-of-the-art algorithms such as the support vector machine on manually selected acoustic features. Furthermore, her model is combined with Lichuan's model to form a hybrid model which produced the best accuracy.

On the implementation side, she sourced necessary software solutions for deploying the classifier. On the Android front-end, she provided starter code to the app and achieved the functionalities of saving speech locally and transcribing speech to text for display. On the back-end, she configured AWS EC2 cloud servers to host the classifier and created a RESTful API using the Flask microframework to handle the prediction requests from mobile clients.

Dovetailing the work on the Android front-end, her contributions allowed the project to achieve an end-to-end system that demonstrates the performance of the developed models.

Lichuan Zhang

The contribution of Lichuan focused on investigating different machine learning approaches that aim to achieve target accuracy as outlined in project requirements. The model implementation was not particularly challenging with existing library, but carefully choosing and tuning hyperparameters required much more effort. Relevant knowledge and experience were crucial when building up a prototype and fine-tuning hyperparameters.

His main contributions include algorithm research, implementation and verification. He experimented and determined potential models and structures that could potentially be employed. Overall, three models were investigated: (1) SVM, (2) CNN+LSTM and (3) final decision network model. These three models utilized both classical and deep learning methods. To achieve this, he read state-of-the-art literature about model implementation. Next, he constructed prototypes for these models. Lastly, he used the RAVNESS dataset to train and fine-tune the model to prevent under-fitting or over-fitting, resulted in higher accuracy.

His work helped the project meet the project constraints: the required 60% accuracy.

Xin Geng

Xin's contribution centred around building the cloud server and the research on the text-based model.

She attempted to use the Google Cloud service at the beginning. Using the Firebase platform, the server and backend implementation progress went well. However, the project met a huge barrier when the team attempted to run the model from Firebase platform. Switching to Google Cloud ML engine was the first backup choice but unfortunately still did not solve the problem. After the investigation and discussion with Kejia Huang, the team decided to use AWS EC2. She configured the mobile client side RESTful API based on the Retrofit framework, which sends the audio to and receives prediction result from the server.

On the algorithm research side, she contributed to the text-based model investigation. She began with reading state-of-the-art literature, then built the LSTM model using features from word2vec libraries. The model was trained on both the positive and negative sentiments and multi-emotion datasets. Her other tasks also include database searching and audio testing.

Although some of her work was not integrated into the final version of the project, these tasks are under the indispensable attempts based on the proposed design. She completed her responsibilities and gained lots of experience in algorithm research and Android application development.

Jiaxin Liu

Jiaxin's contributions focused on the GUI development, along with other tasks in the client-server implementation and research and development. She has implemented all GUI features and back-end functionalities. The GUI was user friendly and visually appealing. The challenge in this task was sorting out the limitations of third-party libraries used in Android and thread management.

Early on when Google Firebase was still the choice of cloud implementation, she wrote modules that communicate between the Android front-end and Firebase. She also implemented the mobile client side and a python local server prior to the RESTful server implementation such that a complete end-to-end prototype can be used to test the model as early as possible. These two tasks were discarded due to updated implementation.

Moreover, she participated in research and development by sourcing an imaging-time-series library that can transcribe audio to an image representation by computing Gramian Angular Summation/Difference Fields and Markov Transition Fields. This is a new literature (2015) and the output images from this technique show visible difference across audio inputs with different emotions. However, currently there is no existing literature that applies this technique to a machine learning algorithm to classify emotions. It is challenging to design and fine-tune the hyperparameters blindly. Hence this feature is not considered as an input to a neural network. She has an interest in employing this technique in the future after gaining more insights into neural networks.

Combining the work on algorithm development, her contributions allowed the project to achieve an end-to-end system that demonstrates the performance of the developed models.

Table of Contents

1.0 Introduction

This report summarizes the motivation, design, implementation, and testing of the project in Automating Speech Emotion Recognition with Machine Learning as part of the final year design project course ECE496. The report concludes with future work.

1.1 Background and Motivation

Speech is the most natural mode of communication employed by humans for "expressing thoughts and emotions through articulate sounds" [1]. Emotions are integral to speech, and the ability to recognize and reason about them from speech signals is critical for enabling natural interactions between humans and machines [2]. These interactions cover a wide spectrum of applications, ranging from clinical monitoring to responsive entertainment systems [3]. Attaining speech-based emotional intelligence will thus greatly empower the next-generation human-computer interfaces in these application areas.

The ability to recognize and classify emotions through speech is the holy grail of emotional artificial intelligence [4]. Defining emotions is a core component of building an automatic speech-based emotion recognizer. Popular research trends in the literature include 1) Discretizing into the archetypal categories of happiness, sadness, fear, anger, surprise, and disgust [5] and 2) Projecting speech into a continuous emotion space indexed by activation and valence [3]. The former approach supports categorization and turns recognition into a classification problem, whereas the latter can capture at finer gradations more subtlety in emotional differences via a regression model [6].

Characterization of speech signals can also play an important role in designing an automatic emotion recognizer. Various characterization methods lead to distinct modeling algorithms, such as Mel-Frequency Cepstral Coefficients (MFCC), Perceptual Linear Prediction (PLP) coefficients, and suprasegmental features, as guided by decades of auditory research [2]. Combined with supervised learning algorithms such as k-nearest neighbours (kNN), hidden Markov model (HMM), and support vector machine (SVM), feature engineering is able to produce robust predictions [2]. Recent works lean more toward a "soft" representation facilitated by the learning capabilities of neural networks and seek end-to-end solutions with deep learning. Convolutional and recurrent neural networks have produced promising results on raw spectral representations of speech [6]. Speech signals can also be seen as the multimodal combination of spoken text and vocal utterances. Hybrid methods that take advantage of information present in multiple modalities are increasingly popular due to their ability to learn models that are applicable across a larger portion of the sample space [7].

In this project, the team proposes to develop a hybrid emotion classification technique that operates CNN and LSTM networks to reason about emotions in speeches. Once trained, the model will output predictions in discrete categories of archetypal emotions. By grasping emotion from both types of networks, the team aims to meaningfully advance the state of the art in speech emotion recognition research.

1.2 Project Goal

The goal of this project is to recognize emotions in speech signals by leveraging machine learning methodologies. In addition, an Android application will be delivered as a graphical user interface to demonstrate the testing results.

1.3 Project Requirements

This section details the requirements of the design.

ID	Project Requirement	Description
1.0	Output: labels of emotion	**Primary functional requirement**: the design shall identify the 6 archetypal emotions: happy, sad, angry, fearful, disgust and surprised [6].
2.0	Input: datasets for training and validation	**Primary functional requirement:** the dataset(s) required to reproduce the results is the Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) [8].
3.0	Input: arbitrary speech signals from users for testing	**Primary functional requirement:** the inputs for testing must be audio speech signals spoken in standard North American English. The inputs must contain sufficient samples for all types of output emotions to conduct a thorough test.
4.0	Maximum length of input speech signals for testing	**Objective**: the design should be able to process up to 5 minutes of speech signals for a single input. **Constraint**: the design shall process at least 5 seconds for a single input.
5.0	Classification accuracy for arbitrary inputs	**Objective:** the design should achieve a true positive rate of 80% [9]. **Constraint:** the design shall achieve at least a true positive rate of 60%.

Table 1.3 Project Requirements

2.0 Final Design

2.1 System-level Overview

The system collects speech signals and displays results via a front-end application, while the recognition is delegated to a classification model hosted on the cloud. The algorithm behind the model builds on existing techniques for emotion recognition.

The process starts with the user recording a speech through an Android application. The application dispatches the recorded audio file through the RESTful API to a server on the cloud. The server triggers the hybrid model, and the model returns the predicted emotion label. The model is an ensemble of acoustics-based (CNN and LSTM) classifiers unified by a final decision neural network which produces a weighted decision from both classifiers. Finally, the Android application will display the identified emotion visually.

The final decision neural network works along with CNN and LSTM model. For the CNN model, the audio signal is transformed to a spectrogram and then fed into an Inception V3 model that is mainly used for image classification. A dense layer is added at the top of the model to perform transfer learning. For the RNN model, raw audio is first passed into a VGG(CNN) network to perform feature extraction. Then the output from VGG is fed into two LSTM layers. The final decision network takes the trained layers from the two models and merges them into a new layer. Then the network is trained again based on the merged layer. The final decision layer is tested to produce better accuracy.

2.2 System Block Diagram

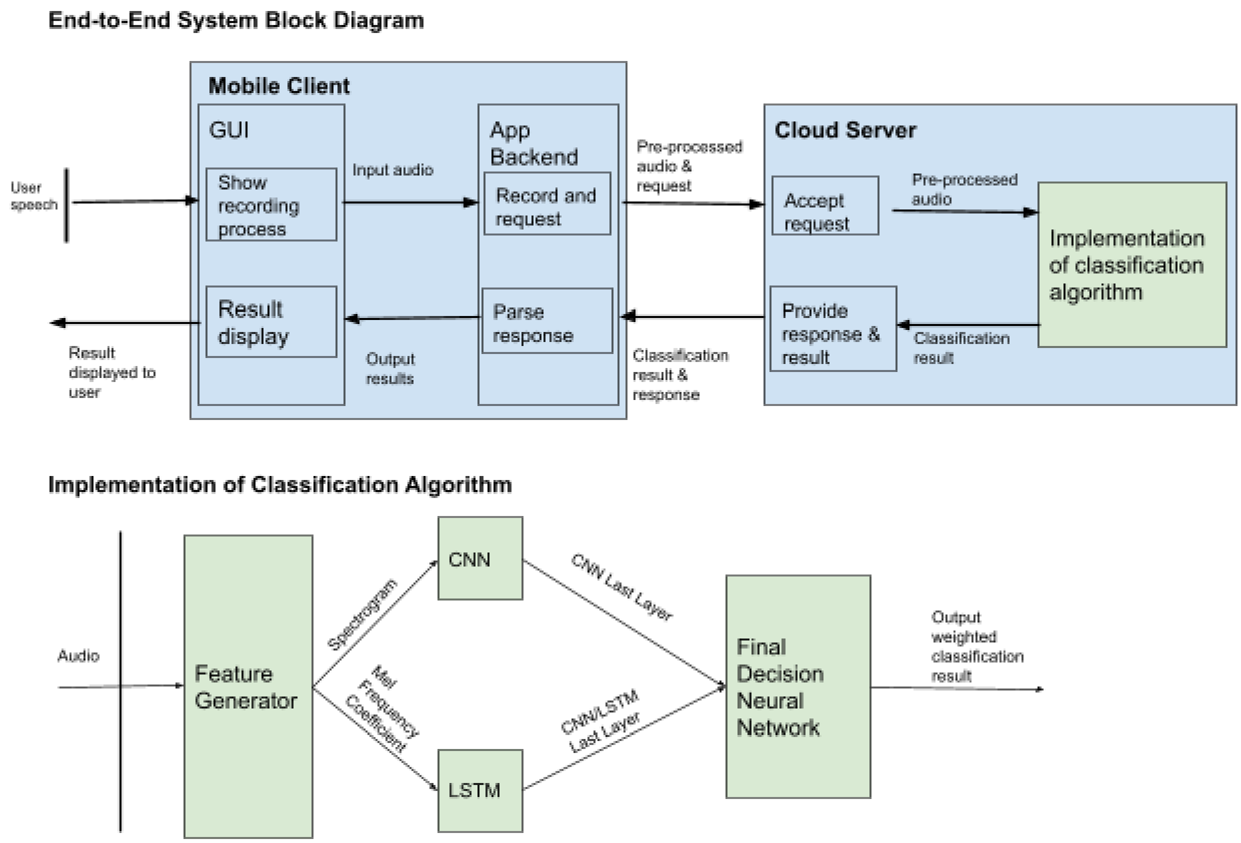


Figure 2.2 System Block Diagram

The project is divided into the following major parts: front-end Android application, back-end server, and research and development in speech to emotion algorithm. The connections between the major parts are shown in the System Block Diagram above.

2.2.1 Front-end Android Application (Jiaxin Liu)

The design and front-end Android Application is completed. The app starts with an entry page displaying the name and the logo of the app “Emotify”. It consists of two fragments Home and New Audio. The Home Page displays the predicted emotion received from the server and the transcribed text. The New Audio Page allows the user to record and play an audio input and sends a request to the server for prediction. The screenshots of the GUI can be found in [Appendix D].

2.2.2 Back-end Implementation (Jiaxin Liu)

The back-end tasks correspond to each item in the front-end. The tasks completed are as follows:

- Records audio in “.wav” format using a third party library and saving in memory
- Uploads audio input through RESTful API client end to AWS EC2 server, and achieves ‘request-response’ communication.
- Transcribes the audio input by sending requests the IBM Watson Speech-to-Text server
- Plays the recorded audio file
- Displays the transcribed text the predicted emotion in emojis
- Pops up windows with messages to inform the user what the app is doing

2.2.3 Server Implementation (Kejia Huang)

To facilitate RESTful API client-server interactions, the classifier is hosted on AWS EC2 cloud server and employed the Flask microframework in Python. The machine learning oriented OS images of AWS EC2 provide ready support for the classifier predicated on Keras and Tensorflow. When the mobile end requests a classification by uploading an audio file, the Flask server invokes the audio preprocessor and the hybrid classifier and returns the emotion label in a JSON object. The scheme of the JSON object is agreed by the server and the client. Upon receiving the server’s response, the mobile client can parse the result and display the identified emotion.

2.2.4 Research and Development in Speech to Emotion Algorithms

2.2.4.1 Spectrograms and CNN (Kejia Huang)

A CNN-based model that operates on the temporal-spectral representation of the speech is proposed and implemented. With eight classes (happiness, sadness, anger, fear, surprise, anxiety, disgust, and neutrality), the model achieved a test accuracy of 55% [Appendix E] on the RAVDESS dataset (SMART Lab, 2014). The spectrogram shows the energy at different frequencies across time for a time-series signal and encodes the acoustic features such as pitch when applied on speech. The spectrogram is thus a suitable visual representation of the speech signal.

Different base architectures have been experimented for the transfer learning and dropout technique are applied to improve the performance of the model. The test accuracy increased to 55% from 42.4% initially, after switching from the MobileNet architecture to Inception v3 and applying a final dropout layer.

2.2.4.2 SVM (Lichuan Zhang)

Traditional machine learning methods (e.g., Support Vector Machine, Hidden Markov Model, etc.) are shown to have similar accuracies with faster processing speed. A prototype is built based on the literature “A new approach in audio emotion recognition” (Ooi et al, 2014). However, from experimentation, these traditional models are not capable of meeting the accuracy goal. Thus this approach is abandoned, and the focus is moved towards deep learning networks.

2.2.4.3 CNN + RNN (Lichuan Zhang)

This model, as its name suggests, is a concatenation of an RNN and a CNN. First spectral features are extracted from the audio waveform in the form of sequential data and fed into multi-layer LSTM(a type of RNN layer) model. The LSTM layers learn the relations between features in the sequence and classify accordingly. The output from the LSTM layers is then channeled to convolutional layers, which employ VGG-like feature extraction with Google Audioset (Hershey et al, 2019; Gemmeke et al, 2019). The combined RNN/CNN model achieves a test accuracy of 54% [Appendix F] on the RAVDESS dataset (SMART Lab, 2014).

2.2.4.4 Text-based LSTM (Xin Geng)

The Text-based LSTM model operated on the tokenized text input, where each word in the text is represented by an integer. In order to classify multiple emotions, the model was trained on Twitter comments dataset [10] (13 emotions in total) first. However, the accuracy on the test data is not promising (10%-26%) [Appendix G]. The second trial was a binary classification between positive and negative sentiments tested on IMDB sentiment dataset and achieved a good accuracy of 85% [Appendix G] [11][12]. Both classifiers were discarded because the training datasets do not synchronize with the 6-8 labels in the requirements, and they produced unsatisfying results.

2.2.4.5 Final Decision Layer (Lichuan Zhang)

The final decision layer takes the last layer of both networks described in 2.2.4.2 and 2.2.4.3, concatenates them and makes a decision based on the combined layer. The purpose of the final decision layer is to predict results based on the feature space from both CNN and RNN networks. By training on the fused layer from two models, the accuracy increased 5%~10% compared to individual models. The final accuracy is 57~60%. [Appendix H]

2.3 Module-level Description and Design

<table>
<tr><th colspan="2">Client User Interface</th></tr>
<tr><td>Input</td><td>1. Audio speech recorded from user
2. Classification result from the server</td></tr>
<tr><td>Output</td><td>1. A request to analyze the audio file
2. Display classification result and transcribed text</td></tr>
<tr><td>Function</td><td>Graphically prompt user for input, playback and display classification result</td></tr>
<tr><th colspan="2">Android Backend</th></tr>
<tr><td>Input</td><td>1. Recorded audio file
2. Incoming classification results from cloud server</td></tr>
<tr><td>Output</td><td>1. Input audio and classification request to be dispatched to server
2. Classification result to be displayed on the user interface</td></tr>
<tr><td>Function</td><td>Respond to user input; bridge between interface and server</td></tr>
<tr><th colspan="2">Cloud Server</th></tr>
<tr><td>Input</td><td>1. Pre-processed audio and request;
2. Result from classification algorithm</td></tr>
<tr><td>Output</td><td>1. Classification results and call to classification algorithm
2. Send respond with classification result</td></tr>
<tr><td>Function</td><td>Data and file storage, algorithm running platform</td></tr>
<tr><th colspan="2">Implementation of Classification Algorithm</th></tr>
<tr><td>Input</td><td>Pre-processed data from the server</td></tr>
<tr><td>Output</td><td>Classification result</td></tr>
<tr><td>Function</td><td>Use multiple machine learning models and produce an ensembled result</td></tr>
</table>

Table 2.3.1 - Modular Description of Android App block Diagram

Feature Generator	
Input	Input data is the pre-processed data received from cloud server
Output	Two split sets of input data, one is in waveform and the other is in text form.
Function	Generate input date for CNN, SVM, and RNN based on the input data they need.
Convolutional Neural Network (CNN)	
Input	Spectrogram of input audio
Output	Probability of each emotion
Function	Use ImageNet neural network to extract and train the emotion features.
Long Short Term Memory(LSTM)	
Input	Features including Mel-frequency cepstral coefficient(MFCC)
Output	Probability of each emotion
Function	Use MFCC feature to train model based on time-sequence data
Final Decision Neural Network	
Input	Probability estimation from CNN and LSTM
Output	Result emotion of the classification
Function	Weight and determine the final output based on input from CNN and RNN models.

Table 2.3.2 - Modular Description of Emotion Classification Algorithm

2.4 Assessment of Final Design

The objective aims to leverage existing machine learning techniques, identify deficiencies, and develop a better technique targeting the multimodal nature of the speech signal with improved prediction accuracy. A limiting factor is the quality and size of training datasets [13]. The text-based approach was discarded due to lack of labeled datasets and its poor accuracy. Therefore, the final design is modified to a hybrid model of CNN and LSTM networks based on the acoustical features only. This hybrid model obtained the best accuracy compared to their individual results as expected. An additional key limitation is that real-time user speech inputs may be different from the training dataset speech samples in many areas. The RAVDESS dataset is recorded in a well-constructed recording environment by professional actors and actresses speaking in the labeled tones. The real-time user input may record noise. The users' emotions may not be as accurate as the professionals. These factors make it challenging to build and label a large dataset of user real-time speech samples, thus cause fluctuation in the accuracy of the model. The team attempts to minimize these impacts at the recording stage to obtain the best accuracy on real-time user inputs.

3.0 Testing and Verification

3.1 Verification table and Validation Matrix

Original Verification Table from proposal in [Appendix C]

ID	Project Requirement	Verification Result and Proof	Requirement Verification Method			
			Similarity	**Review of Design**	**Analysis**	**Test**
1.0	Output: types of emotions to be identified	**Pass.** The final prediction result show on Android app. [Appendix D]		✓		
2.0	Input: training and validation datasets	**Pass**. Using the validation/test API in Keras, an accuracy of 60% accuracy is achieved on randomly selected test set samples.	✓			
3.0	Input: arbitrary speech signals from users for testing	**Pass.** See the transcription display of Android app. [Appendix D]		✓		
4.0	Input: length of signals	**Pass.**				✓
5.0	Emotion recognition accuracy	**Tested.** Test result in [Appendix I]			✓	

Table 3.1 - Updated Validation Matrix

3.2 Final Test Results (system and module-level)

System Level	**Test Results**
Overall functionality: record and send audio input, analyze and receive results	When a mobile client completes recording and requests a prediction, the predicted emotion is displayed through the GUI.
Module-Level	
Client User Interface	Interact successfully with user and display the prediction result in text. [Appendix D]
Android Backend	Record the audio and communicate with cloud server to get the prediction result. [Appendix J]
Cloud Server	Succeed host the analysis model and sent back the prediction result to Android application. [Appendix J]
Implementation of Classification Algorithm	The model with the highest accuracy from models listed below is picked, which is the final decision neural network.
Feature Generator	Transcribe audio to spectrogram and MFCC successfully. [Appendix K]
Convolutional Neural Network	Final accuracy of test data is around 55%. [Appendix E]
Long Short Term Memory	Final accuracy of test data is around 54 %. [Appendix F]
Final Decision Neural Network	Final accuracy of test data is around 57%-60%. [Appendix H]

Table 3.2 - Final Test Results

4.0 Summary and Conclusions

Humans express thoughts and emotions naturally through speech and articulated sounds. Recognizing emotions through speech is critical for enabling natural interactions between humans and machines. As such, the goal of this project is to categorize emotions from arbitrary speeches conducted in standard North American English. To solve this problem, first, the output space of emotions is defined as archetypal categories including happy, sad, angry, fearful, disgust and surprised. Next, a hybrid methodology is proposed and implemented leveraging machine learning techniques including CNN and RNN networks. The text-based model has been investigated but discarded due to poor accuracy and lack of labeled datasets. An Android front-end application is delivered enabling an end-to-end pipeline. Based on relevant assessments and the progress of the project, the objective is to reach a true positive rate of 60%. This project is completed and concludes with the delivery of the hybrid CNN and RNN model, an end-to-end pipeline that allows the users to test and interact with the model. Future work involves a showcase activity with a displaying poster at the design fair in early April.

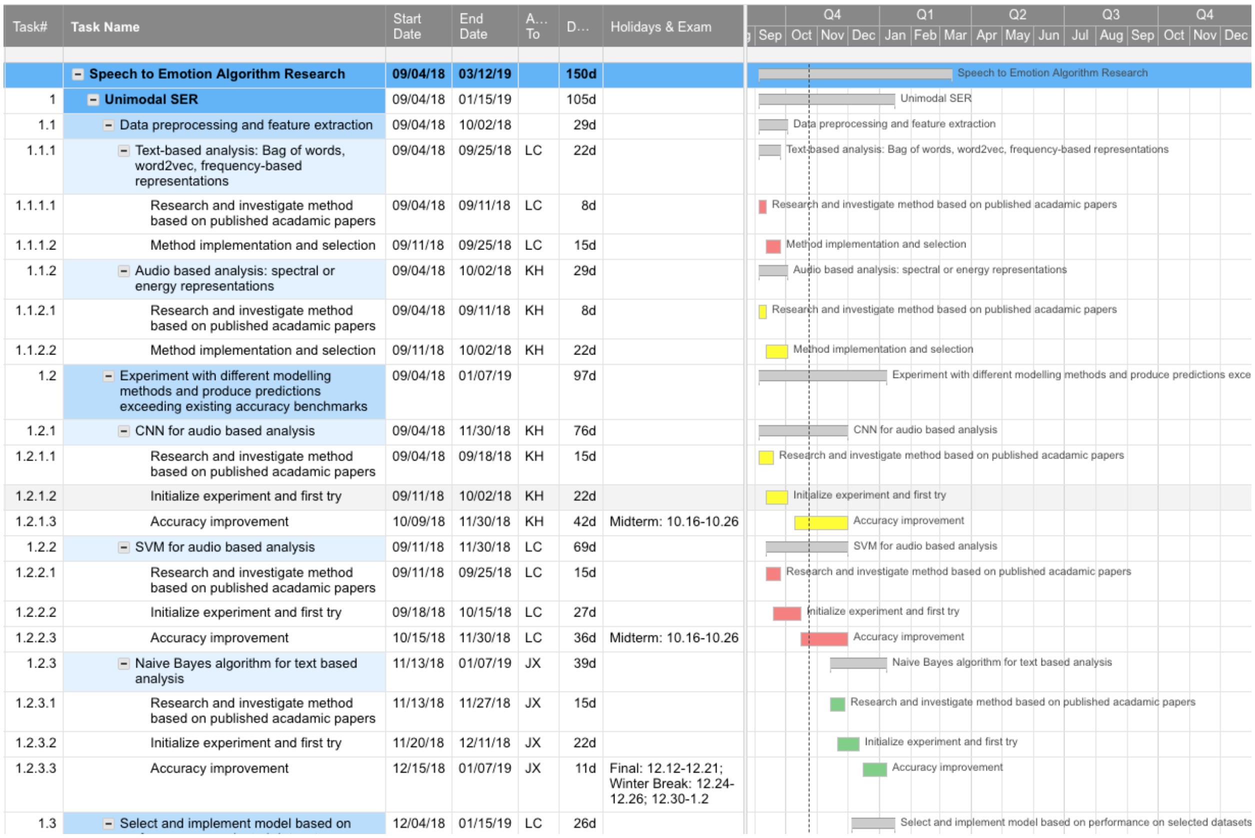
5.0 References

[1] M. E. Ayadi, M. S. Kamel, and F. Karray, “Survey on speech emotion recognition: Features, classification schemes, and databases,” *Pattern Recognition*, vol. 44, no. 3, pp. 572–587, 2011.
[2] P. Tzirakis, J. Zhang, and B. W. Schuller, “End-to-End Speech Emotion Recognition Using Deep Neural Networks,” *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.
[3] R. Cowie, E. Douglas-Cowie, N. Tsapatsoulis, G. Votsis, S. Kollias, W. Fellenz, and J. Taylor, “Emotion recognition in human-computer interaction,” *IEEE Signal Processing Magazine*, vol. 18, no. 1, pp. 32–80, 2001.
[4] Z. Ivcevic, M. A. Brackett, and J. D. Mayer, “Emotional Intelligence and Emotional Creativity,” *Journal of Personality*, Apr. 2007.
[5] P. Ekman, “An argument for basic emotions,” *Cognition and Emotion*, vol. 6, no. 3-4, pp. 169–200, 1992.
[6] B. W. Schuller, “Speech emotion recognition,” *Communications of the ACM*, vol. 61, no. 5, pp. 90–99, 2018.
[7] Z. Xie and L. Guan, “Multimodal Information Fusion of Audio Emotion Recognition
[8] Livingstone SR, Russo FA (2018) The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English. PLoS ONE 13(5): e0196391. https://doi.org/10.1371/journal.pone.0196391
[9]Ooi, C., Seng, K., Ang, L. and Chew, L. (2014). A new approach of audio emotion recognition. *Expert Systems with Applications*, 41(13), pp.5858-5869.
[10] Sentiment Analysis in Text Dataset, Sentiment Analysis: Emotion in Text, Crowdflower, 2016, https://data.world/crowdflower/sentiment-analysis-in-text
[11] Sentiment analysis in IMDB movie reviews dataset, Determine whether a movie review is positive or negative,2016, https://www.kaggle.com/c/sentiment-analysis-on-imdb-movie-reviews
[12] Large Movie Review Dataset, Stanford University, 2011, http://ai.stanford.edu/~amaas/data/sentiment/
[13] El Ayadi, M., Kamel, M. and Karray, F. (2011). Survey on speech emotion recognition: Features, classification schemes, and databases. *Pattern Recognition*, 44(3), pp.572-587.

6.0 Appendices

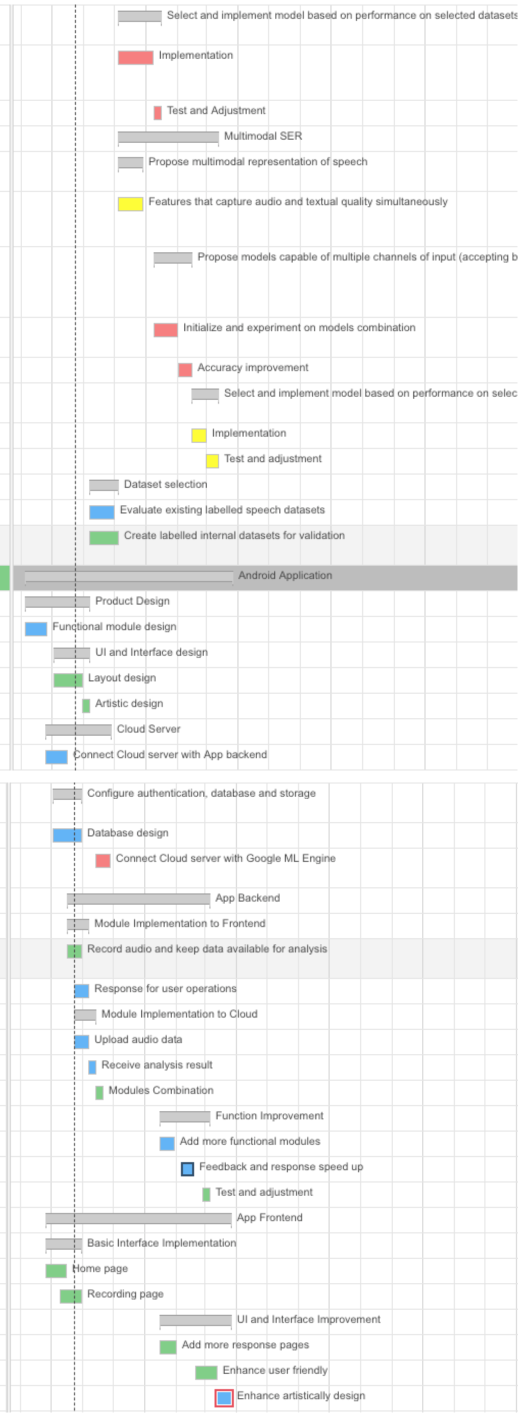
Appendix A: Gantt Chart History

Gantt Chart in Proposal:



Task#	Task Name	Start Date	End Date	A… To	D…	Holidays & Exam
	Speech to Emotion Algorithm Research	**09/04/18**	**03/12/19**		**150d**	
1	**Unimodal SER**	09/04/18	01/15/19		105d	
1.1	Data preprocessing and feature extraction	09/04/18	10/02/18		29d	
1.1.1	Text-based analysis: Bag of words, word2vec, frequency-based representations	09/04/18	09/25/18	LC	22d	
1.1.1.1	Research and investigate method based on published acadamic papers	09/04/18	09/11/18	LC	8d	
1.1.1.2	Method implementation and selection	09/11/18	09/25/18	LC	15d	
1.1.2	Audio based analysis: spectral or energy representations	09/04/18	10/02/18	KH	29d	
1.1.2.1	Research and investigate method based on published acadamic papers	09/04/18	09/11/18	KH	8d	
1.1.2.2	Method implementation and selection	09/11/18	10/02/18	KH	22d	
1.2	Experiment with different modelling methods and produce predictions exceeding existing accuracy benchmarks	09/04/18	01/07/19		97d	
1.2.1	CNN for audio based analysis	09/04/18	11/30/18	KH	76d	
1.2.1.1	Research and investigate method based on published acadamic papers	09/04/18	09/18/18	KH	15d	
1.2.1.2	Initialize experiment and first try	09/11/18	10/02/18	KH	22d	
1.2.1.3	Accuracy improvement	10/09/18	11/30/18	KH	42d	Midterm: 10.16-10.26
1.2.2	SVM for audio based analysis	09/11/18	11/30/18	LC	69d	
1.2.2.1	Research and investigate method based on published acadamic papers	09/11/18	09/25/18	LC	15d	
1.2.2.2	Initialize experiment and first try	09/18/18	10/15/18	LC	27d	
1.2.2.3	Accuracy improvement	10/15/18	11/30/18	LC	36d	Midterm: 10.16-10.26
1.2.3	Naive Bayes algorithm for text based analysis	11/13/18	01/07/19	JX	39d	
1.2.3.1	Research and investigate method based on published acadamic papers	11/13/18	11/27/18	JX	15d	
1.2.3.2	Initialize experiment and first try	11/20/18	12/11/18	JX	22d	
1.2.3.3	Accuracy improvement	12/15/18	01/07/19	JX	11d	Final: 12.12-12.21; Winter Break: 12.24-12.26; 12.30-1.2
1.3	Select and implement model based on	12/04/18	01/15/19	LC	26d	

1.3	Select and implement model based on performance on selected datasets	12/04/18	01/15/19	LC	26d	
1.3.1	Implementation	12/04/18	01/07/19	LC	18d	Final: 12.12-12.21; Winter Break: 12.24-12.26; 12.30-1.2
1.3.2	Test and Adjustment	01/08/19	01/15/19	LC	8d	
2	**Multimodal SER**	12/04/18	03/12/19		71d	
2.1	Propose multimodal representation of speech	12/04/18	12/28/18	KH	12d	
2.1.1	Features that capture audio and textual quality simultaneously	12/04/18	12/28/18	KH	12d	Final: 12.12-12.21; Winter Break: 12.24-12.26;
2.2	Propose models capable of multiple channels of input (accepting both audio and text) that produce predictions exceeding existing accuracy benchmarks	01/08/19	02/14/19	LC	36d	
2.2.1	Initialize and experiment on models combination	01/08/19	01/31/19	LC	24d	
2.2.2	Accuracy improvement	02/01/19	02/14/19	LC	12d	Family Day: 2.11-2.12
2.3	Select and implement model based on performance on selected datasets	02/14/19	03/12/19	KH	18d	
2.3.1	Implementation	02/14/19	02/28/19	KH	6d	Midterm: 2.18-2.26
2.3.2	Test and adjustment	02/28/19	03/12/19	KH	13d	
3	**Dataset selection**	11/06/18	12/04/18		29d	
3.1	Evaluate existing labelled speech datasets	11/06/18	11/30/18	XG	25d	
3.2	Create labelled internal datasets for validation	11/06/18	12/04/18	JX	29d	
	Android Application	**09/04/18**	**03/26/19**		**164d**	
4	**Product Design**	09/04/18	11/06/18		52d	
4.1	Functional module design	09/04/18	09/25/18	XG	22d	
4.2	UI and Interface design	10/02/18	11/06/18		24d	
4.2.1	Layout design	10/02/18	10/30/18	JX	17d	Midterm: 10.16-10.26
4.2.2	Artistic design	10/30/18	11/06/18	JX	8d	
5	**Cloud Server**	09/24/18	11/27/18		53d	
5.1	Connect Cloud server with App backend	09/24/18	10/15/18	XG	21d	
5.2	Configure authentication, database and storage	10/02/18	10/30/18	XG	17d	
5.2.1	Database design	10/02/18	10/30/18	XG	17d	Midterm: 10.16-10.26
5.3	Connect Cloud server with Google ML Engine	11/13/18	11/27/18	LC	15d	
6	**App Backend**	10/16/18	03/05/19		103d	
6.1	Module Implementation to Frontend	10/16/18	11/06/18		12d	
6.1.1	Record audio and keep data available for analysis	10/16/18	10/30/18	JX	5d	
6.1.2	Response for user operations	10/23/18	11/06/18	XG	12d	
6.2	Module Implementation to Cloud	10/23/18	11/13/18		19d	
6.2.1	Upload audio data	10/23/18	11/06/18	XG	12d	
6.2.2	Receive analysis result	11/06/18	11/13/18	XG	8d	
6.3	Modules Combination	11/13/18	11/20/18	JX	8d	
6.4	Function Improvement	01/15/19	03/05/19		39d	
6.4.1	Add more functional modules	01/15/19	01/29/19	XG	15d	
6.4.2	Feedback and response speed up	02/05/19	02/17/19	XG	11d	Family Day: 2.11-2.12
6.4.3	Test and adjustment	02/26/19	03/05/19	JX	8d	
7	**App Frontend**	09/25/18	03/26/19		143d	
7.1	Basic Interface Implementation	09/25/18	10/30/18		24d	
7.1.1	Home page	09/25/18	10/15/18	JX	20d	
7.1.2	Recording page	10/09/18	10/30/18	JX	11d	Midterm: 10.16-10.26
7.2	UI and Interface Improvement	01/15/19	03/26/19		60d	
7.2.1	Add more response pages	01/15/19	01/31/19	JX	17d	
7.2.2	Enhance user friendly	02/19/19	03/12/19	JX	15d	Midterm: 2.18-2.26
7.2.3	Enhance artistically design	03/12/19	03/26/19	XG	15d	



Gantt Chart in Progress Update:

GANTT CHART TEMPLATE

Smartsheet Tip • A Gantt chart's visual timeline allows you to see details about each task as well as project dependencies.

PROJECT TITLE Measuring Conversations Using Machine Learning 2: Measuring Conversations

COMPANY NAME ECE496

Exam and Holiday Session: Midterm: 10.16-10.26; Final: 12.12-12.21; Winter Break: 12.24-12.26; 12.30-1.2; Family Day: 2.11-2.12; Midterm: 2.18-2.26

TASK#	TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION (Day)	PCT OF TASK COMPLETE
	Speech to Emotion Algorithm Research		**09/04/18**	**03/26/19**	**203**	50%
1	**Unimodal SER**		09/04/18	02/28/19	177	80%
1.1	Data preprocessing and feature extraction		09/04/18	10/02/18	28	0%
1.1.1	Text-based analysis: Bag of words, word2vec, frequency-based representations	LC	09/04/18	09/30/18	26	0%
1.1.1.1	Research and investigate architecture based on published academic papers	LC	09/04/18	09/11/18	7	100%
1.1.1.2	Architecture selection	LC	09/11/18	09/25/18	14	100%
1.1.1.3	Sentiment Analysis Algorithm	LC	09/25/18	09/30/18	5	100%
1.1.2	Audio based analysis: spectral or energy representations	KH	09/04/18	10/02/18	28	85%
1.1.2.1	Research and investigate method based on published acadamic papers	KH	09/04/18	09/11/18	7	70%
1.1.2.2	Method implementation and selection	KH	09/11/18	10/02/18	21	100%
1.2	Experiment with different modelling methods and produce predictions exceeding existing accuracy benchmarks		09/04/18	02/14/19	163	100%
1.2.1	CNN for audio based analysis	KH	09/04/18	11/30/18	87	100%
1.2.1.1	Research and investigate method based on published acadamic papers	KH	09/04/18	09/18/18	14	100%
1.2.1.2	Initialize experiment and first try	KH	09/11/18	10/02/18	21	100%
1.2.1.3	Accuracy improvement	KH	10/09/18	11/30/18	52	100%
1.2.2	SVM for audio based analysis	LC	09/11/18	11/30/18	80	100%
1.2.2.1	Research and investigate method based on published acadamic papers	LC	09/11/18	09/25/18	14	100%
1.2.2.2	Initialize experiment and first try	LC	09/18/18	10/15/18	27	100%
1.2.3	LSTM for text based analysis (word2vec)	XG	12/27/18	02/14/19	49	30%
1.2.3.1	Research and investigate method based on published acadamic papers	XG	12/27/18	01/31/19	35	50%
1.2.3.2	Initialize experiment and first try	XG	01/02/19	01/31/19	29	70%
1.2.3.3	Accuracy improvement	XG	01/31/19	02/14/19	14	10%
1.2.4	CNN for audio based analysis (time series to image)	JX	12/20/18	02/14/19	56	30%
1.2.4.1	Research and investigate method based on published acadamic papers	JX	12/20/18	01/11/19	22	100%
1.2.4.2	Initialize experiment and first try	JX	01/14/19	01/31/19	17	30%
1.2.4.3	Accuracy improvement	JX	01/31/19	02/14/19	14	0%
1.3	Select and implement model based on performance on selected datasets		01/31/19	02/28/19	28	0%
1.3.1	Implementation	LC	01/31/19	02/15/19	15	0%
1.3.2	Test and Adjustment	LC	01/31/19	02/28/19	28	0%
1.4	CNN+RNN prototype method		10/29/18	01/20/19	83	80%
1.4.1	CNN+RNN prototype Investigation	LC	10/29/18	11/18/18	20	100%
1.4.2	CNN+RNN prototype build	LC	11/18/18	12/06/18	18	100%
1.4.3	Accuracy improvement	LC	12/06/18	01/20/19	45	30%
2	**Multimodal SER**		01/20/19	03/26/19	65	0%
2.1	Propose multimodal representation of speech	KH	01/20/19	01/31/19	11	0%
2.1.1	Features that capture audio and textual quality simultaneously	KH	01/20/19	01/31/19	11	0%
2.2	Propose models capable of multiple channels of input (accepting both audio and text) that produce predictions exceeding existing accuracy benchmarks	LC	01/31/19	02/28/19	28	0%
2.2.1	Initialize and experiment on models combination	LC	01/31/19	02/20/19	20	0%
2.2.2	Accuracy improvement	LC	02/20/19	02/28/19	8	0%
2.3	Select and implement model based on performance on selected datasets	KH	03/01/19	03/26/19	25	0%
2.3.1	Implementation	KH	03/01/19	03/10/19	9	0%

2018 Fall Term				2019 Winter Term		
September	October	November	December	Januray	Febrary	March
4 11 18 25	2 9 16 23 30	6 13 20 27	4 11 18 25	3 10 17 24 31	7 14 21 28	7 14 21 28

2.3.2	Test and adjustment	KH	03/10/19	03/26/19	16	0%
3	**Dataset selection**		11/20/18	03/26/19	126	0%
3.1	Evaluate existing labelled speech datasets	XG	11/20/18	11/30/18	10	90%
3.2	Create labelled internal datasets for validation	JX	02/26/19	03/26/19	28	0%
	Android Application		**09/04/18**	**03/26/19**	203	70%
4	**Product Design**		09/04/18	11/06/18	63	100%
4.1	Functional module design	XG	09/04/18	09/25/18	21	100%
4.2	UI and Interface design		10/02/18	11/06/18	35	100%
4.2.1	Layout design/Artistic design	JX	10/02/18	10/30/18	28	100%
5	**Cloud Server**		09/24/18	01/30/19	128	80%
5.1	Connect Cloud server with App backend	XG	09/24/18	10/15/18	21	100%
5.2	Configure database and storage	XG	10/02/18	10/30/18	28	100%
5.2.1	Database design	XG	10/02/18	10/30/18	28	100%
5.3	Cloud Function	XG	10/30/18	11/27/18	28	100%
5.4	Model Training on the Cloud	KH	12/21/18	1/5/19	15	100%
5.4.1	Standardize input data I/O format for running model training on Google ML Engine	KH	12/21/18	12/31/18	10	100%
5.4.2	Develop scripts to orchestrate and automate model training with Google ML Engine	KH	01/02/19	1/5/19	3	100%
5.5	Deploy Model on the Cloud	KH	01/05/19	01/30/19	25	40%
5.5.1	Standardize model format for deployment and prediction on Google ML Engine	KH	01/05/19	01/20/19	15	60%
5.5.2	Develop scripts to invoke prediction and collection results	KH	01/20/19	01/30/19	10	20%
6	**App Backend**		10/16/18	03/05/19	140	60%
6.1	Module Implementation to Frontend		10/16/18	11/30/18	45	70%
6.1.1	Record audio and keep data available for playback and analysis	JX	11/06/18	11/30/18	24	100%
6.1.2	Transcribe speech to text	JX	11/06/18	11/30/18	24	100%
6.2	Module Implementation to Cloud		11/06/18	02/14/19	100	50%
6.2.1	Upload audio data	JX	11/06/18	11/30/18	24	100%
6.2.2	Receive analysis result	XG	11/24/18	02/14/19	82	30%
6.3	Modules Combination	JX	02/14/19	02/27/19	13	0%
6.4	Function Improvement		02/01/19	03/05/19	32	0%
6.4.1	Add more functional modules	XG	02/01/19	02/14/19	13	0%
6.4.2	Feedback and response speed up	XG	02/05/19	02/17/19	12	0%
6.4.3	Test and adjustment	JX	02/26/19	03/05/19	7	0%
7	**App Frontend**		09/25/18	03/26/19	182	0%
7.1	Basic Interface Implementation		09/25/18	10/30/18	35	100%
7.1.1	Home page	JX	09/25/18	10/15/18	20	100%
7.1.2	Recording page	JX	10/09/18	11/30/18	52	100%
7.2	UI and Interface Improvement		02/19/19	03/26/19	35	0%
7.2.1	Enhance user friendly	JX	02/19/19	03/12/19	21	0%
7.2.2	Enhance artistically design	XG	03/12/19	03/26/19	14	0%

Gantt Chart in Final Design:

Comment for the cells in **yellow**: after investigation and trials, these parts of tasks were not succeeded, removed or changed to other plans for final design at last.

PROJECT TITLE	Measuring Conversations Using Machine Learning 2: Measuring Conversations	COMPANY NAME	ECE496

Exam and Holiday Session: *Midterm: 10.16-10.26; Final: 12.12-12.21; Winter Break: 12.24-12.26; 12.30-1.2; Family Day: 2.11-2.12; Midterm: 2.18-2.26*

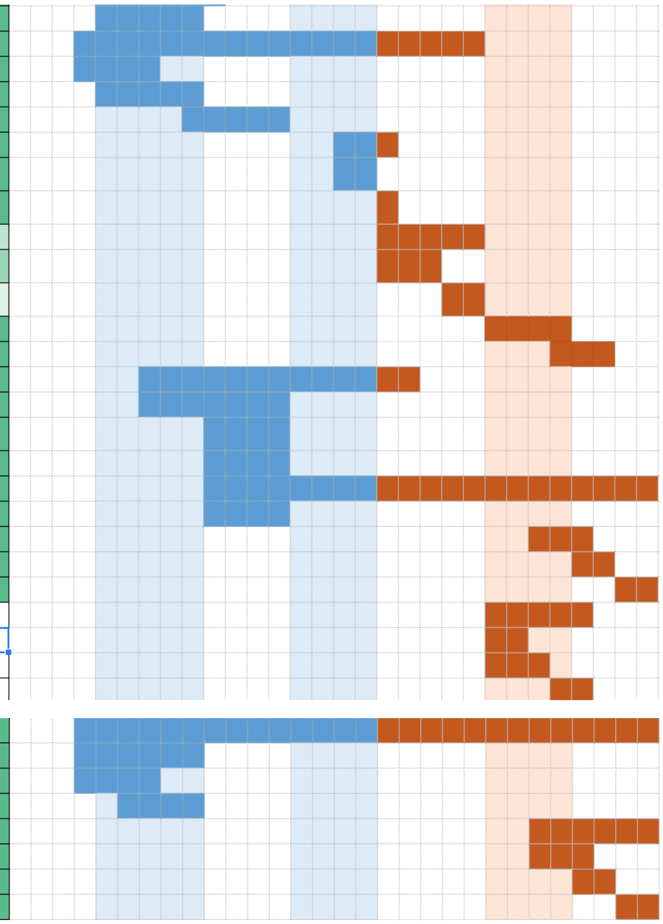
Comment for the cells in yellow: *after investigation and trials, these parts of tasks were not successed, removed or changed to other plans for final design at last.*

2018 Fall Term: September (4, 11, 18, 25), October (2, 9, 16, 23, 30), November (6, 13, 20, 27), December (4, 11, 18, 25)

2019 Winter Term: Januray (3, 10, 17, 24, 31), Febrary (7, 14, 21, 28), March (7, 14, 21, 28)

TASK#	TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION (Day)	PCT OF TASK COMPLETE
	Speech to Emotion Algorithm Research		**09/04/18**	**03/21/19**	**198**	100%
1	**Unimodal SER**		09/04/18	02/28/19	**177**	100%
1.1	Data preprocessing and feature extraction		09/04/18	10/02/18	**28**	100%
1.1.1	Text-based analysis: Bag of words, word2vec, frequency-based representations	LC	09/04/18	09/30/18	**26**	100%
1.1.1.1	Research and investigate architecture based on published academic papers	LC	09/04/18	09/11/18	**7**	100%
1.1.1.2	Architecture selection	LC	09/11/18	09/25/18	**14**	100%
1.1.1.3	Sentiment Analysis Algorithm	LC	09/25/18	09/30/18	**5**	100%
1.1.2	Audio based analysis: spectral or energy representations	KH	09/04/18	10/02/18	**28**	100%
1.1.2.1	Research and investigate method based on published acadamic papers	KH	09/04/18	09/11/18	**7**	100%
1.1.2.2	Method implementation and selection	KH	09/11/18	10/02/18	**21**	100%
1.2	Experiment with different modelling methods and produce predictions exceeding existing accuracy benchmarks		09/04/18	02/14/19	**163**	100%
1.2.1	CNN for audio based analysis	KH	09/04/18	11/30/18	**87**	100%
1.2.1.1	Research and investigate method based on published acadamic papers	KH	09/04/18	09/18/18	**14**	100%
1.2.1.2	Initialize experiment and first try	KH	09/11/18	10/02/18	**21**	100%
1.2.1.3	Accuracy improvement	KH	10/09/18	11/30/18	**52**	100%
1.2.2	SVM for audio based analysis	LC	09/11/18	11/30/18	**80**	100%
1.2.2.1	Research and investigate method based on published acadamic papers	LC	09/11/18	09/25/18	**14**	100%
1.2.2.2	Initialize experiment and first try	LC	09/18/18	10/15/18	**27**	100%
1.2.3	LSTM for text based analysis (word2vec)	XG	12/27/18	02/14/19	**49**	100%
1.2.3.1	Research and investigate method based on published acadamic papers	XG	12/27/18	01/31/19	**35**	100%
1.2.3.2	Initialize experiment and first try	XG	01/02/19	01/31/19	**29**	100%
1.2.3.3	Accuracy improvement	XG	01/31/19	02/14/19	**14**	100%
~~1.2.4~~	~~CNN for audio based analysis (time series to image)~~	~~JX~~	~~12/20/18~~	~~02/14/19~~	~~**56**~~	~~30%~~
~~1.2.4.1~~	~~Research and investigate method based on published academic papers~~	~~JX~~	~~12/20/18~~	~~01/11/19~~	~~**22**~~	~~100%~~
~~1.2.4.2~~	~~Initialize experiment and first try~~	~~JX~~	~~01/14/19~~	~~01/31/19~~	~~**17**~~	~~30%~~
~~1.2.4.3~~	~~Accuracy improvement~~	~~JX~~	~~01/31/19~~	~~02/14/19~~	~~**14**~~	~~0%~~
1.3	Select and implement model based on performance on selected datasets		01/31/19	02/28/19	**28**	100%
1.3.1	Implementation	LC	01/31/19	02/15/19	**15**	100%
1.3.2	Test and Adjustment	LC	01/31/19	02/28/19	**28**	100%
1.4	CNN+RNN prototype method		10/29/18	01/20/19	**83**	100%
1.4.1	CNN+RNN prototype Investigation	LC	10/29/18	11/18/18	**20**	100%
1.4.2	CNN+RNN prototype build	LC	11/18/18	12/06/18	**18**	100%
1.4.3	Accuracy improvement	LC	12/06/18	01/20/19	**45**	100%
2	**Multimodal SER**		01/20/19	03/21/19	**60**	100%
2.1	Propose multimodal representation of speech, integrate CNN prototype(1.2.1) and CNN+RNN prototype (1.4) together	LC	01/20/19	01/31/19	**11**	100%
2.1.1	Initialize and experiment on models combination	LC	01/31/19	02/14/19	**14**	100%
2.1.2	Accuracy improvement	LC	02/28/19	03/14/19	**14**	100%
2.2	Test and adjustment	KH	03/10/19	03/21/19	**11**	100%
3	**Dataset selection**		11/20/18	03/21/19	**121**	100%
3.1	Evaluate existing labelled speech datasets	XG	11/20/18	11/30/18	**10**	100%
3.2	Create labelled real person audio datasets for test	XG	03/14/19	03/21/19	**7**	100%
	Android Application		**09/04/18**	**03/21/19**	**198**	100%
4	**Product Design**		09/04/18	11/06/18	**63**	100%
4.1	Functional module design	XG	09/04/18	09/25/18	**21**	100%
4.2	UI and Interface design		10/02/18	11/06/18	**35**	100%

4.2.1	Layout design/Artistic design	JX	10/02/18	10/30/18	**28**	100%
5	**Cloud Server**		09/24/18	01/30/19	**128**	100%
5.1	Connect Cloud server with App backend	XG	09/24/18	10/15/18	**21**	100%
5.2	Configure database and storage	XG	10/02/18	10/30/18	**28**	100%
5.3	Cloud Function	XG	10/30/18	11/27/18	**28**	100%
5.4	Model Training on the AWS EC2 Cloud Server	KH	12/21/18	1/5/19	**15**	100%
~~5.4.1~~	~~Standardize input data I/O format for running model training on Google ML Engine~~	~~KH~~	~~12/21/18~~	~~12/31/18~~	~~**10**~~	~~100%~~
~~5.4.2~~	~~Develop scripts to orchestrate and automate model training with Google ML Engine~~	~~KH~~	~~01/02/19~~	~~1/5/19~~	~~**3**~~	~~100%~~
~~5.5~~	~~Deploy Model on the Google Cloud Platform~~	~~KH~~	~~01/05/19~~	~~01/30/19~~	~~**25**~~	~~40%~~
~~5.5.1~~	~~Standardize model format for deployment and prediction on Google ML Engine~~	~~KH~~	~~01/05/19~~	~~01/20/19~~	~~**15**~~	~~60%~~
~~5.5.2~~	~~Develop scripts to invoke prediction and collection results~~	~~KH~~	~~01/20/19~~	~~01/30/19~~	~~**10**~~	~~20%~~
5.6	Config the AWS EC2 Cloud Server	KH	02/05/19	02/28/19	**23**	100%
5.6.1	Implementation the Flask Python server	KH	02/28/19	03/14/19	**14**	100%
6	**App Backend**		10/16/18	03/05/19	**140**	100%
6.1	Module Implementation to Frontend		10/16/18	11/30/18	**45**	100%
6.1.1	Record audio and keep data available for playback and analysis	JX	11/06/18	11/30/18	**24**	100%
6.1.2	Transcribe speech to text	JX	11/06/18	11/30/18	**24**	100%
6.2	Module Implementation to Cloud		11/06/18	03/21/19	**135**	100%
6.2.0	Previous Google Cloud Configeration	XG	11/06/18	11/30/18	**24**	100%
6.2.1	Upload audio to EC2 server by Retrofit framework	XG	02/14/19	03/07/19	**21**	100%
6.2.2	Receive analysis result from EC2 server	XG	03/07/19	03/12/19	**5**	100%
6.3	Modules Combination	JX	03/12/19	03/21/19	**9**	100%
~~6.4~~	~~Function Improvement~~		~~02/01/19~~	~~03/05/19~~	~~**32**~~	~~0%~~
~~6.4.1~~	~~Add more functional modules~~	~~XG~~	~~02/01/19~~	~~02/14/19~~	~~**13**~~	~~0%~~
~~6.4.2~~	~~Feedback and response speed up~~	~~XG~~	~~02/05/19~~	~~02/17/19~~	~~**12**~~	~~0%~~
~~6.4.3~~	~~Test and adjustment~~	~~JX~~	~~02/26/19~~	~~03/05/19~~	~~**7**~~	~~0%~~
7	**App Frontend**		09/25/18	03/21/19	**177**	100%
7.1	Basic Interface Implementation		09/25/18	10/30/18	**35**	100%
7.1.1	Home page	JX	09/25/18	10/15/18	**20**	100%
7.1.2	Recording page	JX	10/09/18	11/30/18	**52**	100%
7.2	UI and Interface Improvement		02/14/19	03/21/19	**35**	100%
7.2.1	Enhance user friendly	JX	02/14/19	03/07/19	**21**	100%
7.2.2	Add pop up user operation reminder	JX	03/07/19	03/14/19	**7**	100%
7.2.3	Enhance artistically design	JX	03/14/19	03/21/19	**7**	100%



Appendix B: Financial Plan

Consumable/Services						
Item	***Priority***	***Cost/Unit***	***Quantity (#or hours)***	***Total Cost***	***Requires Funding***	
Internet Data Plan	1	$100/mo	4 x 10% x 8 mo	$320	N	
Total Consumables/Services				$320		
Total Requiring Funding				$0		
Capital Equipment **Comment: Yellow Cell Items are removed from final design**						

Item	*Priority*	*Cost/Unit*	*Quantity (#or hours)*	*Total Cost*	*Requires Funding*	*Kept/Paid for by Student*
AWS Platform	1	$17/month	3	$51		$51
~~Firebase - Cloud Function, CPU/second~~	~~1~~	~~$0.01/thou -sand~~	~~538~~	~~$5.38~~	~~N~~	~~$5.38~~
~~Google Cloud - ML Engine~~	~~1~~	~~$0.28~~	~~20~~	~~$5.73~~	~~N~~	~~$5.73~~
Development System	2	$6,000	10%	$600	N	N
Android Cell Phone	2	$500	10%	$50.00	N	N
Total Capital Equipment				$701.00		
Total Requiring Funding				$0		
Student Labour						
Item	*Cost/Unit*	*Quantity (#or hours)*	*Total Cost*			
Student1	$25	200	$5,000			
Student2	$25	200	$5,000			
Student3	$25	200	$5,000			
Student4	$25	200	$5,000			
Total Student Labour (unfunded)			$20,000			
Summary				**Funding**		
Total Cost of Project			$21,021.00	Students ($100 ea)		$400
Total Cost Requiring Funding			$0	Supervisor		$0
				Request from Design Centre		N/A
				Total Funding		$400

Appendix C: Validation and Acceptance Tests

This section details of tests need to validate that the proposed solution meets the requirements.

ID	Project Requirement	Acceptance Tests
1.0	Output: types of emotions to be identified	**Review of design**: The output is accepted if it's one of the six labels defined in requirements.
2.0	Input: training and validation datasets	**Similarity**: The RAVDESS is validated and widely used in research of emotion recognition through speech [8].
3.0	Input: arbitrary speech signals from users for testing	**Review of design:** The input is accepted if it is an audio file spoken in standard North American English by arbitrary speakers.
4.0	Input: length of signals	**Test**: Direct measurement
5.0	Emotion recognition accuracy	**Analysis**: number of correct output labels / number of total test cases The final solution will be tested to identify emotions from 30 sample inputs (5 samples for each emotion). Each speaker will provide one or two samples and also label the emotions themselves.

Verification Table

Appendix D: Android Application GUI

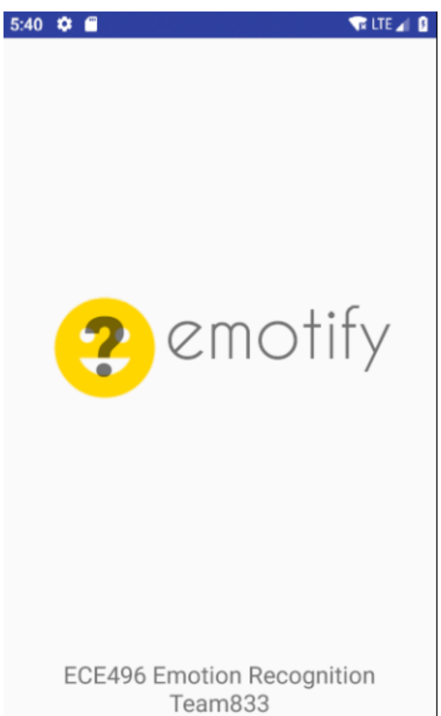


Figure 1: Welcome page

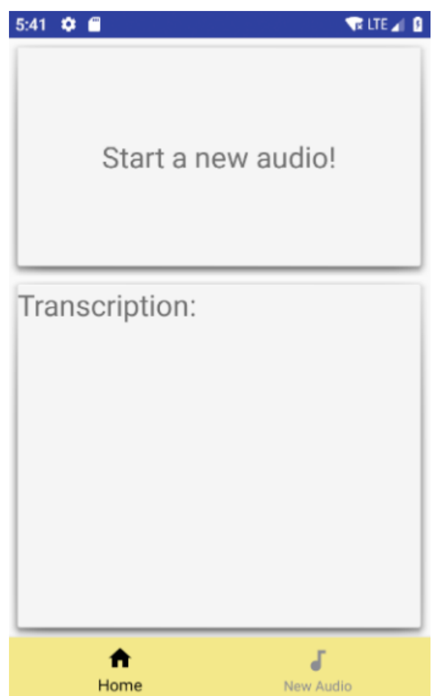


Figure 2: Home Page

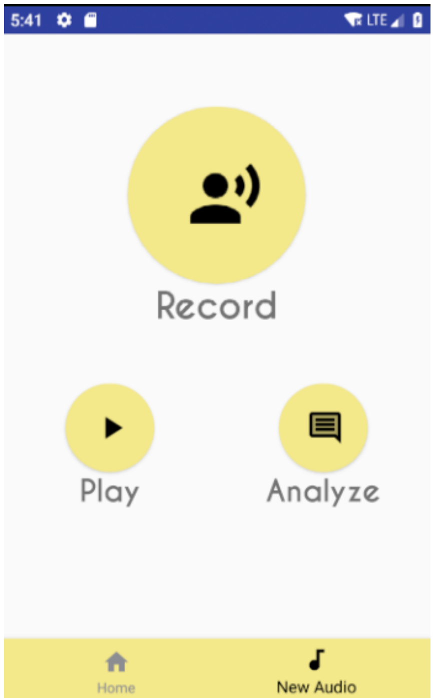


Figure 3: New Audio Page

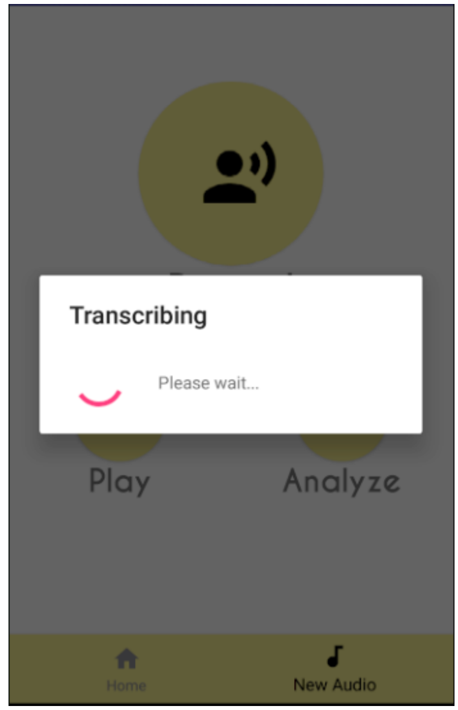


Figure 4: Transcribing popup window

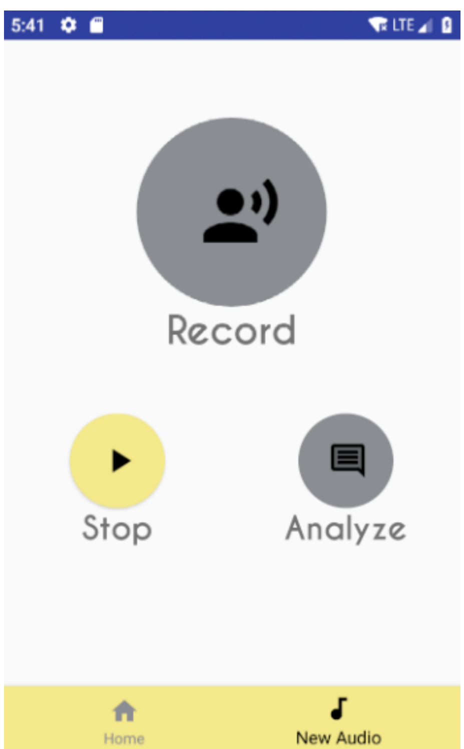


Figure 5: Playing the audio recorded

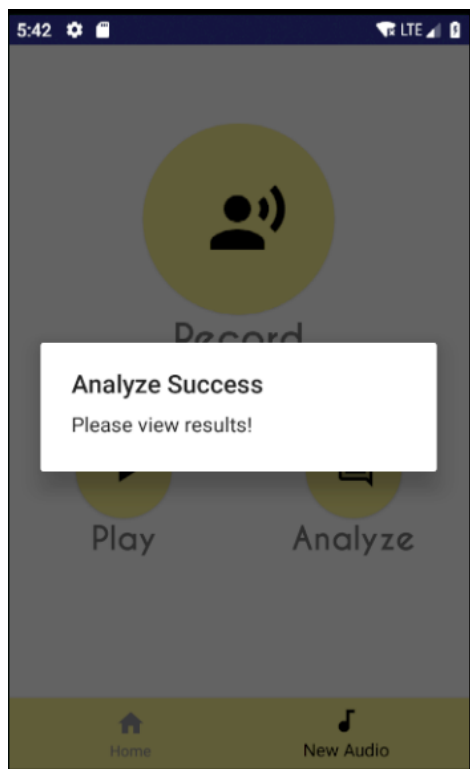


Figure 6: Analysis popup window

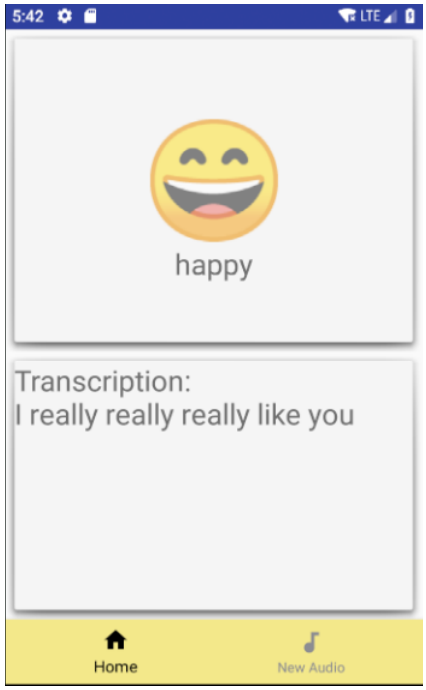


Figure 7: Home Page with result

Appendix E: Convolutional Neural Network

Note: model not tuned for maximum validation accuracy

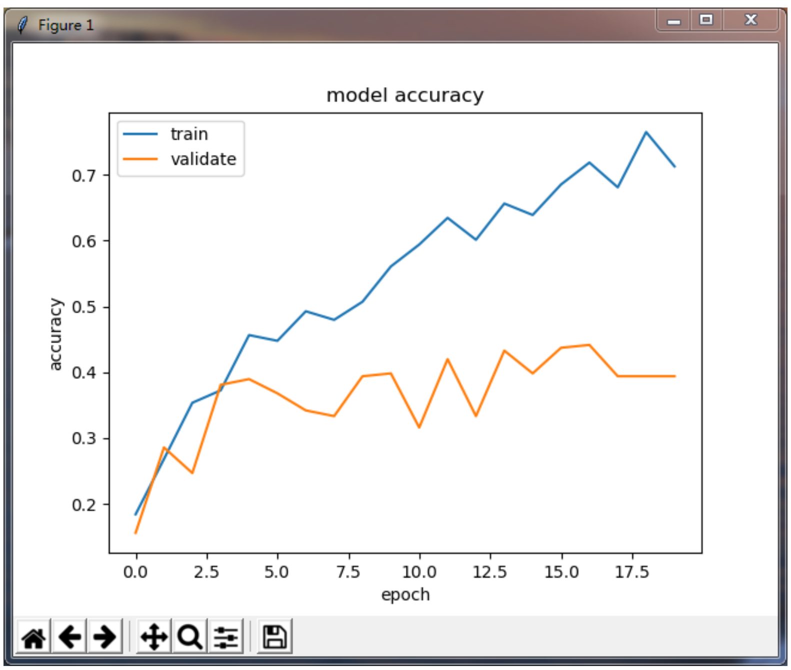


Figure 1: Model accuracy plot for train and validate data

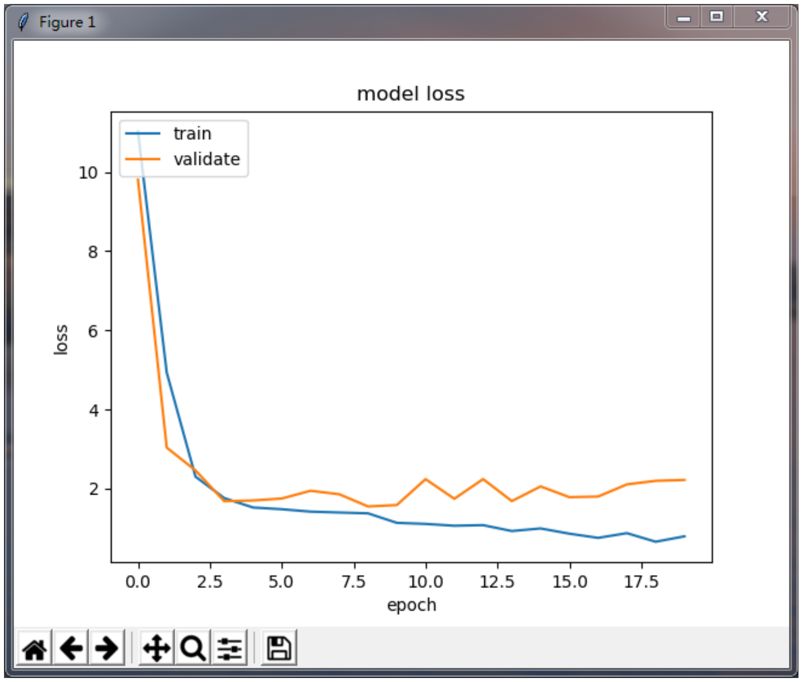


Figure 2: Model loss plot for train and validate data

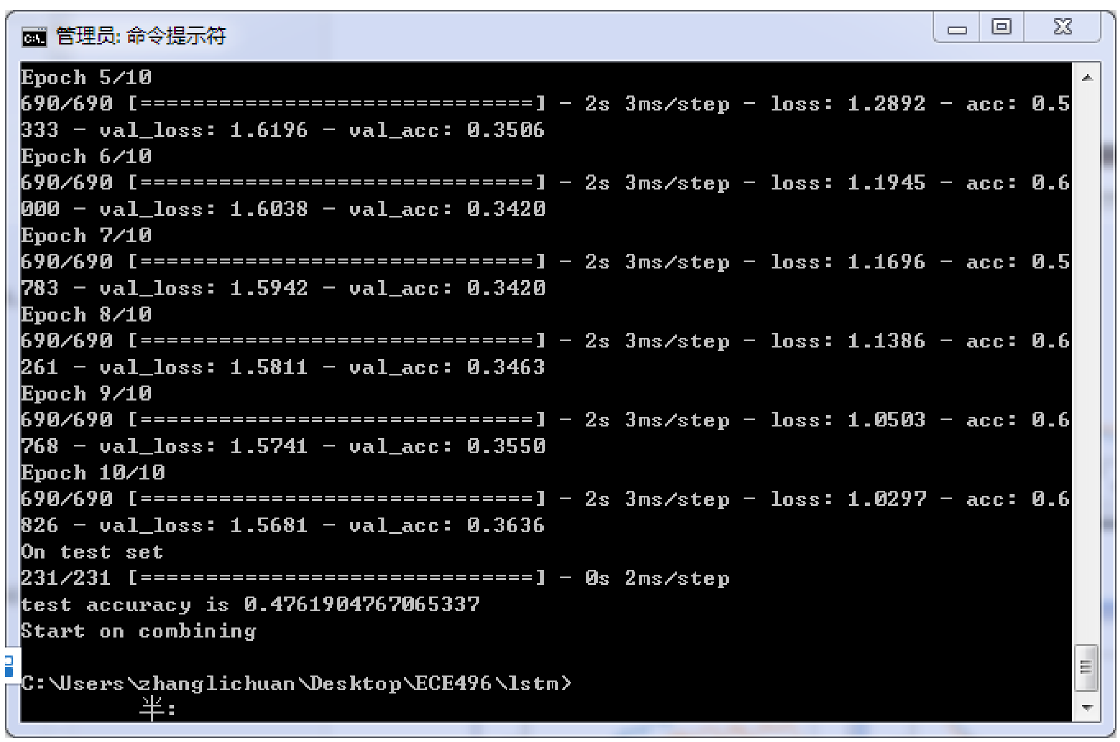


Figure 3: Test data accuracy

Appendix F: LSTM Model for Audio-based analysis

Note: model not tuned for maximum validation accuracy

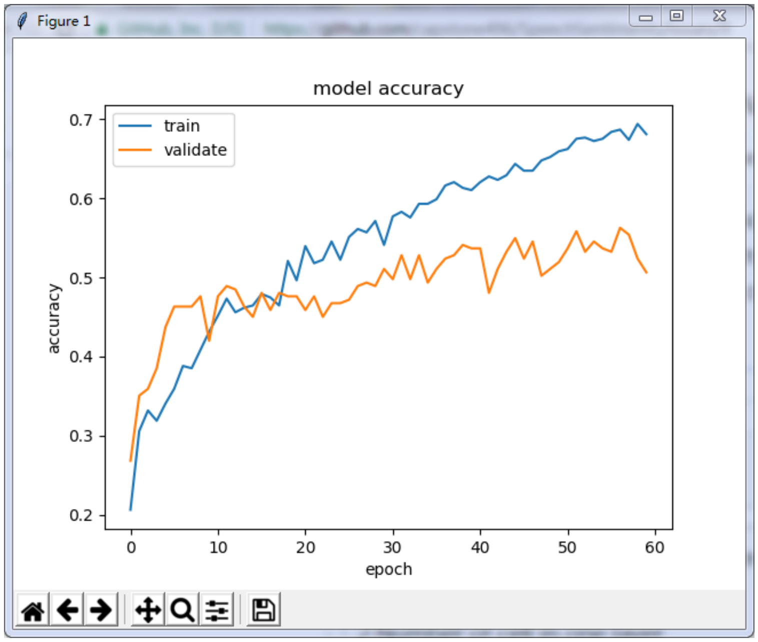


Figure 1: Model accuracy plot for train and validate data

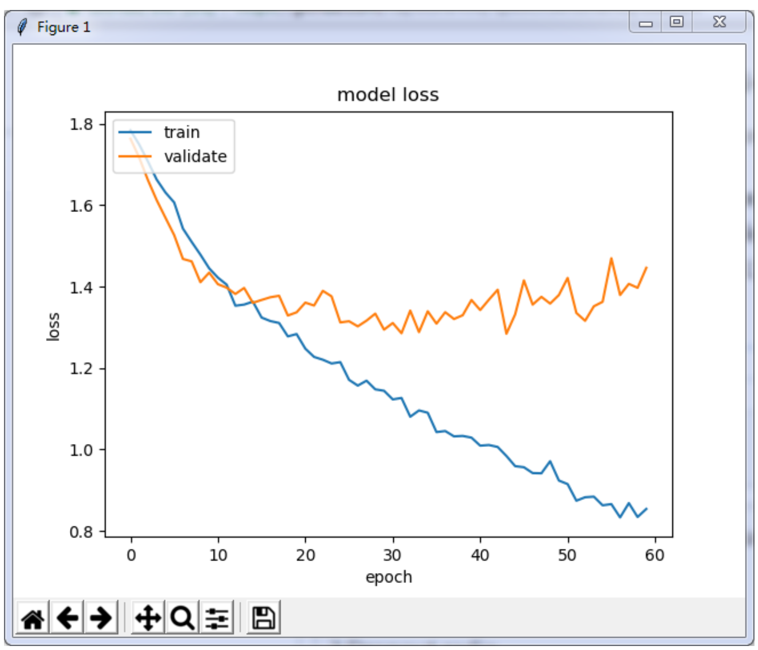
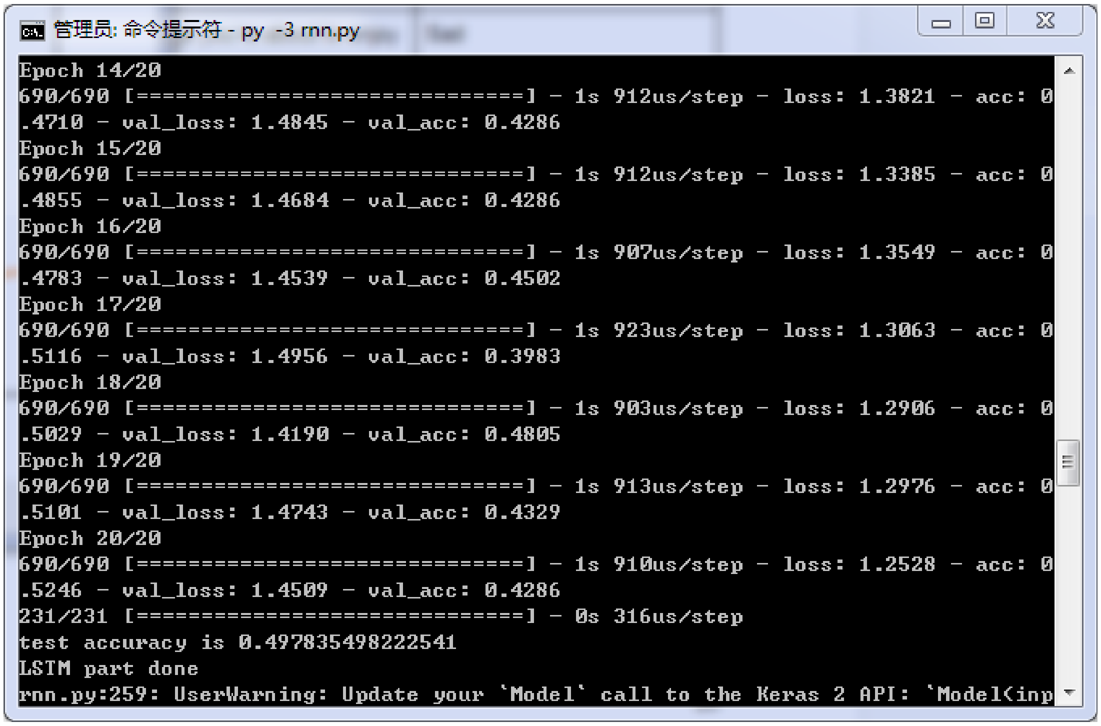


Figure 2: Model loss plot for train and validate data




```
Epoch 14/20
690/690 [==============================] - 1s 912us/step - loss: 1.3821 - acc: 0
.4710 - val_loss: 1.4845 - val_acc: 0.4286
Epoch 15/20
690/690 [==============================] - 1s 912us/step - loss: 1.3385 - acc: 0
.4855 - val_loss: 1.4684 - val_acc: 0.4286
Epoch 16/20
690/690 [==============================] - 1s 907us/step - loss: 1.3549 - acc: 0
.4783 - val_loss: 1.4539 - val_acc: 0.4502
Epoch 17/20
690/690 [==============================] - 1s 923us/step - loss: 1.3063 - acc: 0
.5116 - val_loss: 1.4956 - val_acc: 0.3983
Epoch 18/20
690/690 [==============================] - 1s 903us/step - loss: 1.2906 - acc: 0
.5029 - val_loss: 1.4190 - val_acc: 0.4805
Epoch 19/20
690/690 [==============================] - 1s 913us/step - loss: 1.2976 - acc: 0
.5101 - val_loss: 1.4743 - val_acc: 0.4329
Epoch 20/20
690/690 [==============================] - 1s 910us/step - loss: 1.2528 - acc: 0
.5246 - val_loss: 1.4509 - val_acc: 0.4286
231/231 [==============================] - 0s 316us/step
test accuracy is 0.497835498222541
LSTM part done
rnn.py:259: UserWarning: Update your `Model` call to the Keras 2 API: `Model(inp
```

Figure 3: Test data accuracy

Appendix G: LSTM Model for Text-based analysis

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_5 (Embedding)      (None, 300, 128)          38400
_________________________________________________________________
spatial_dropout1d_5 (Spatial (None, 300, 128)          0
_________________________________________________________________
lstm_5 (LSTM)                (None, 64)                49408
_________________________________________________________________
dense_5 (Dense)              (None, 13)                845
=================================================================
Total params: 88,653
Trainable params: 88,653
Non-trainable params: 0
_________________________________________________________________
None
Train on 22500 samples, validate on 7500 samples
Epoch 1/6
22500/22500 [==============================] - 146s 7ms/step - loss: 2.1070 - acc: 0.2802 - val_loss:
2.4656 - val_acc: 0.1227
Epoch 2/6
22500/22500 [==============================] - 143s 6ms/step - loss: 2.0200 - acc: 0.2895 - val_loss:
2.4806 - val_acc: 0.1227
Epoch 3/6
22500/22500 [==============================] - 137s 6ms/step - loss: 2.0202 - acc: 0.2895 - val_loss:
2.4734 - val_acc: 0.1227
Epoch 4/6
22500/22500 [==============================] - 137s 6ms/step - loss: 2.0197 - acc: 0.2895 - val_loss:
2.4718 - val_acc: 0.1227
Epoch 5/6
22500/22500 [==============================] - 143s 6ms/step - loss: 2.0193 - acc: 0.2895 - val_loss:
2.4752 - val_acc: 0.1227
Epoch 6/6
22500/22500 [==============================] - 125s 6ms/step - loss: 2.0197 - acc: 0.2895 - val_loss:
2.4289 - val_acc: 0.1227
10000/10000 [==============================] - 25s 3ms/step
Accuracy: 10.26%
```

Figure 1: 13 emotion labels test data accuracy for LSTM model

```
Review 0 of 10000
Review 1000 of 10000
Review 2000 of 10000
Review 3000 of 10000
Review 4000 of 10000
Review 5000 of 10000
Review 6000 of 10000
Review 7000 of 10000
Review 8000 of 10000
Review 9000 of 10000
Fitting random forest to training data....
train accuracy:  0.9637333333333333
test accuracy:  0.266
```

Figure 2: Higher test data (13 labels) accuracy by using random forest

```
Total params: 81,961
Trainable params: 81,961
Non-trainable params: 0
_________________________________________________________________
Train on 23750 samples, validate on 1250 samples
Epoch 1/3
23750/23750 [==============================] - 1129s 48ms/step - loss: 0.6716 - acc: 0.5685
- val_loss: 0.6209 - val_acc: 0.6520
Epoch 2/3
23750/23750 [==============================] - 1104s 47ms/step - loss: 0.4455 - acc: 0.7959
- val_loss: 0.3585 - val_acc: 0.8536
Epoch 3/3
23750/23750 [==============================] - 1043s 44ms/step - loss: 0.2961 - acc: 0.8807
- val_loss: 0.3502 - val_acc: 0.8560
25000/25000 [==============================] - 460s 18ms/step
Accuracy: 85.14%
```

Figure 3: Sentiment test data accuracy

Appendix H: Final Decision Network

Note: model not tuned for maximum validation accuracy

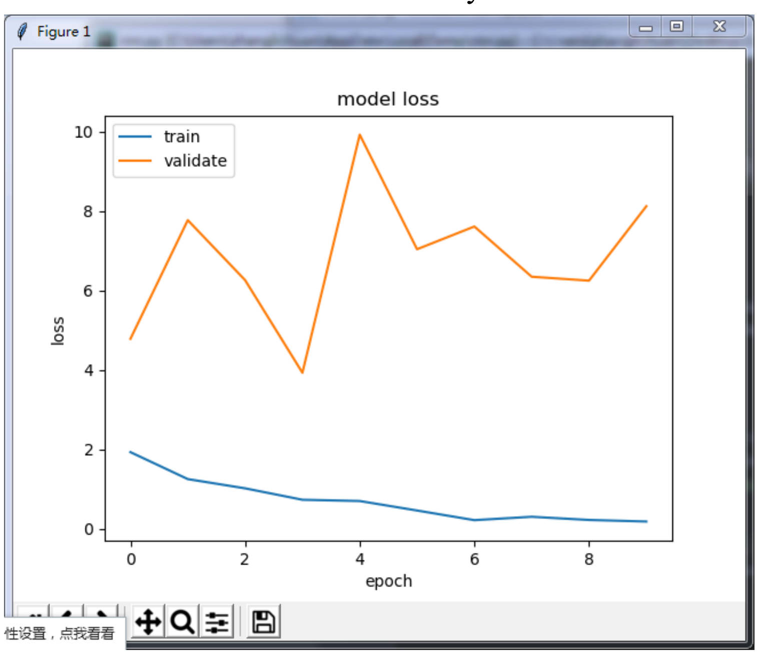


Figure 1: Model accuracy plot for train and validate data

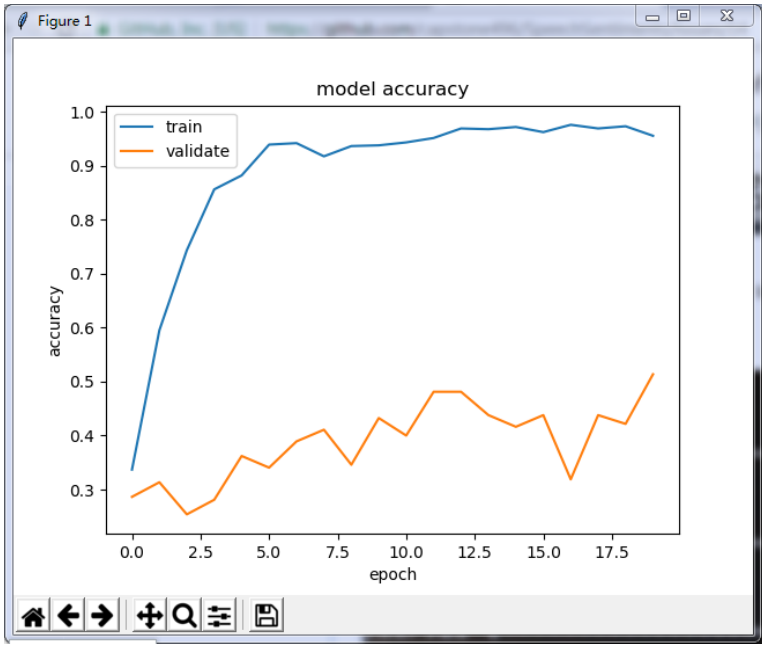


Figure 2: Model loss plot for train and validate data

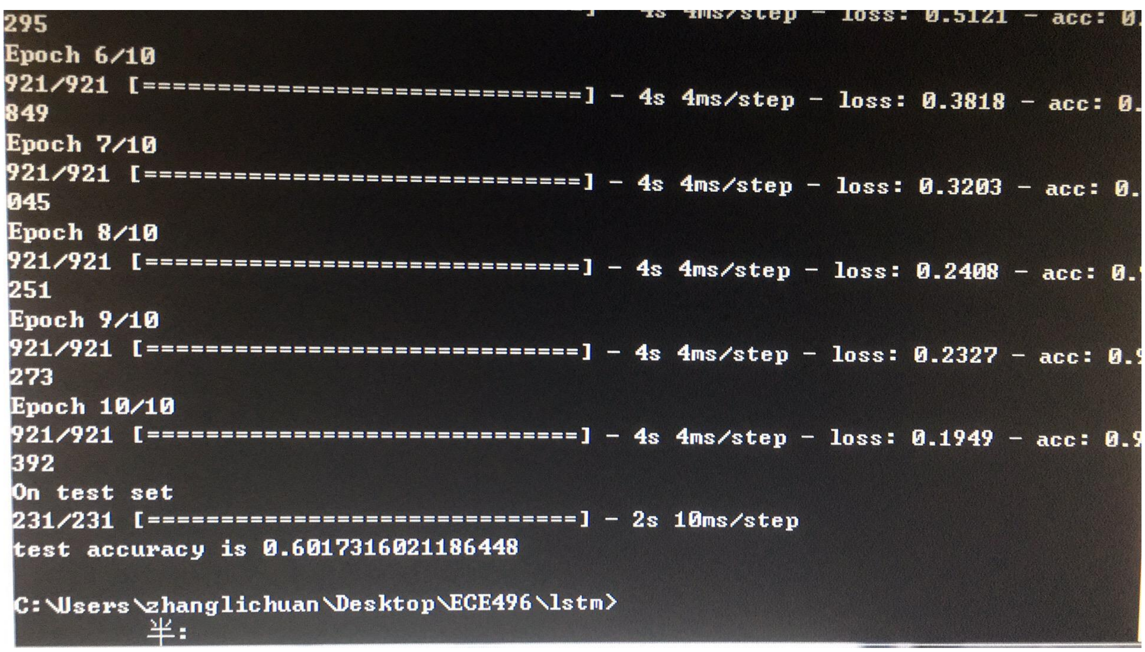


Figure 3: Test data accuracy

Appendix I: Real person audio test

Ground Truth Emotion	**Speech Transcription**	**Predicted Emotion**
Happy	I’m so happy!	Disgust
	Hey, I won a lottery.	Happy
	Wow, this party is so great!	Happy
	What wonderful weather.	Happy
	Hahaha, it’s so funny.	Happy
Angry	What is it?	Happy
	Holy, my dog ruined my new bag.	Angry
	Blow, winds, and crack your cheeks! Rage! Blow!	Disgust
Sad	I’m so sad.	Angry
	This movie really moved me.	Happy
	The city is sacked.	Happy
	I, that never weep, but now I woe	Disgust
Disgust	Ew, what the hell is it?	Happy
	Thou elvish-mark’d, abortive, rooting hog.	Sad
	We waste good surprises on you.	Sad
Surprised	OMG, I love it so much, thank you.	Happy
	Wow, here is a pretty girl.	Happy
	Oh, Juliet you are so beautiful today.	Happy
Fearful	A line from Jack in *Titanic*	sad

Appendix J: Android Backend and Server communication

```
03-21 10:54:03.756 5087-5087/project.ece496.emotionrecogspeechgui E/AudioRecordTest: calling comm in record
03-21 10:54:05.931 5087-5087/project.ece496.emotionrecogspeechgui I/CredentialUtils: JNDI string lookups is not available.
03-21 10:54:05.932 5087-5087/project.ece496.emotionrecogspeechgui I/CredentialUtils: JNDI string lookups is not available.
03-21 10:54:05.956 5087-5087/project.ece496.emotionrecogspeechgui D/NetworkSecurityConfig: No Network Security Config specified, using platform default
03-21 10:54:05.975 5087-5087/project.ece496.emotionrecogspeechgui E/AudioRecordTest: calling comm in record
03-21 10:54:06.053 5087-5092/project.ece496.emotionrecogspeechgui I/zygote: Do partial code cache collection, code=123KB, data=78KB
    After code cache collection, code=123KB, data=78KB
    Increasing code cache capacity to 512KB
```

Figure 1: Backend log for successful record

```
    Content-Length: 185472
    RIFF�� ��WAVEfmt ����������D������X�� �� ��data�� ����������������������������������������
    ����������������������������������������������������������������������������������������
03-21 10:54:51.287 5087-5487/project.ece496.emotionrecogspeechgui I/chatty: uid=10080(project.ece496.emotionrecogspeechgui) .75.47:7000
03-21 10:54:51.287 5087-5487/project.ece496.emotionrecogspeechgui D/OkHttp: ��������������������������������������������������������
03-21 10:54:54.079 5087-5487/project.ece496.emotionrecogspeechgui D/OkHttp: <-- 200 OK http://18.224.75.47:7000/api/predict (2791ms)
    Content-Type: application/json
    Content-Length: 33
    Server: Werkzeug/0.14.1 Python/3.6.6
03-21 10:54:54.080 5087-5487/project.ece496.emotionrecogspeechgui D/OkHttp: Date: Thu, 21 Mar 2019 14:54:56 GMT
03-21 10:54:54.084 5087-5487/project.ece496.emotionrecogspeechgui D/OkHttp: {"predictedEmotion":"surprised"}
    <-- END HTTP (33-byte body)
03-21 10:54:54.093 5087-5087/project.ece496.emotionrecogspeechgui D/MainActivity: getting a response...
03-21 10:54:54.094 5087-5087/project.ece496.emotionrecogspeechgui E/MainActivity: Result surprised
```

Figure 2: Backend log for upload and receive analysis result

```
(tensorflow_p36) ubuntu@ip-172-31-11-90:~/khuang/PythonServer$ python flask_server.py
2019-03-21 14:50:51.981445: W tensorflow/core/framework/op_def_util.cc:346] Op BatchNormWithGlobalNormalization is deprecated. It will cease to work in GraphDef version 9. Use tf.nn.batch_normalization().
 * Serving Flask app "flask_server" (lazy loading)
 * Environment: production
   WARNING: Do not use the development server in a production environment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://0.0.0.0:7000/ (Press CTRL+C to quit)
2019-03-21 14:54:54.791587: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX512F
2019-03-21 14:54:54.793284: I tensorflow/core/common_runtime/process_util.cc:69] Creating new thread pool with default inter op setting: 2. Tune using inter_op_parallelism_threads for best performance.
138.51.240.151 - - [21/Mar/2019 14:54:56] "POST /api/predict HTTP/1.1" 200 -
```

```
(tensorflow_p36) ubuntu@ip-172-31-11-90:~/khuang/PythonServer/received_audio$ ls -l
total 1452
-rw-rw-r-- 1 ubuntu ubuntu 508078 Mar 14 13:22 03-01-02-02-02-01-05.wav
-rw-rw-r-- 1 ubuntu ubuntu 210284 Mar 13 18:36 03-01-05-01-01-01-01.wav
-rw-rw-r-- 1 ubuntu ubuntu 185472 Mar 21 14:54 cache3a6c4243-c605-4930-a127-4d8e1b49c41c
```

Figure 3: Server logs for successfully receiving and storing audio file

Appendix K: Feature Generator

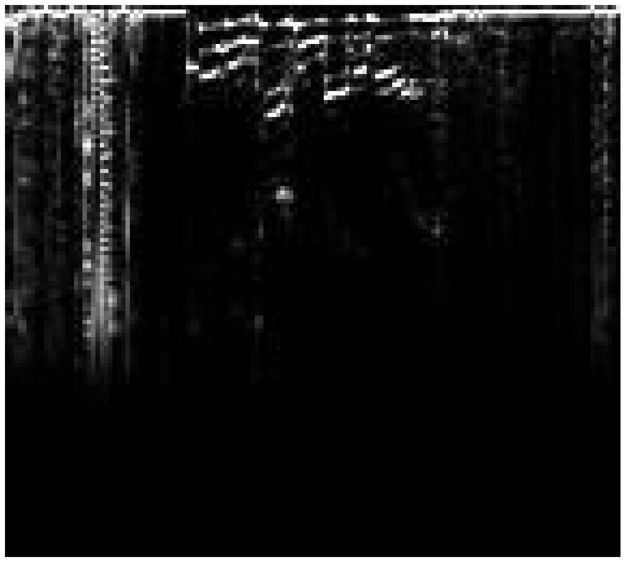


Figure 1: Spectrogram generate from samples of input

Group Final Report Attribution Table

This table should be filled out to accurately reflect who contributed to each section of the report and what they contributed. Provide a **column** for each student, a **row** for each major section of the report, and the appropriate codes (e.g. 'RD, MR') in each of the necessary **cells** in the table. You may expand the table, inserting rows as needed, but you should not require more than two pages. The original completed and signed form must be included in the hard copies of the final report. Please make a copy of it for your own reference.

Section	Student Initials			
	LC	KH	JX	XG
Executive Summary	ET	ET	RD, MR	
Acknowledgement	ET	ET	ET	RD, MR
Group Contribution	ET	ET	ET	RD, MR
Individual Contribution	RD, MR	RD,M R	RD, MR, ET	RD, MR
1.1 Background and Motivation		RD, MR	ET	
1.2 Project Goal			RD, MR	
1.3 Project Requirements			RS, RD, MR	
2.1 System-level Overview	MR, ET	RD, MR		
2.2.1 Front-end Android			RS, RD, MR	

2.2.2 Back-end Implementation		ET	RS, RD, MR	ET
2.2.3 Server Implementation		RS, RD, MR		ET
2.2.4.1 Spectrograms and CNN	ET	RS, RD, MR		
2.2.4.2 SVM	RS, RD, MR		ET	
2.2.4.3 CNN+RNN	RS, RD, MR	ET	ET	
2.2.4.4 LSTM for Text	ET		ET	RS, RD, MR
2.2.4.5 Final Decision Layer	RS, RD, MR	ET	ET	
2.3 Module-level Descriptions	RS, RD, MR	ET		ET
2.4 Assessment of Proposed Solution	ET		RS, RD, MR	
3.1 Verification table	ET		RS, RD	RD, MR, ET
3.2 Final Test Results	ET	ET		RD, MR,

				ET
Conclusion	ET		RD, MR	
Reference	ET		RD	ET
Appendix	RD, ET	RD, ET	RD, ET	RD, MR, ET
All	FP, CM	FP, CM	FP, CM	FP, CM

Abbreviation Codes:

Fill in abbreviations for roles for each of the required content elements. You do not have to fill in every cell. The "**All**" row refers to the complete report and should indicate who was responsible for the final compilation and final read through of the completed document.

RS – responsible for research of information
RD – wrote the first draft
MR – responsible for major revision
ET – edited for grammar, spelling, and expression
OR – other
"All" row abbreviations:
FP – final read through of complete document for flow and consistency
CM – responsible for compiling the elements into the complete document
OR - other
If you put OR (other) in a cell please put it in as OR1, OR2, etc. Explain briefly below the role referred to:
OR1: enter brief description here
OR2: enter brief description here

Signatures

By signing below, you verify that you have read the attribution table and agree that it accurately reflects your contribution to this document.

Name ______________ **Signature** ______________ **Date:**

Name ______________ **Signature** ______________ **Date:**

Name ______________ **Signature** ______________ **Date:**

Name ______________ **Signature** ______________ **Date:**