Voice Emotion Recognition with Deep Learning

Group: A+++++
Group ID: 03

Victoria Meng (Xuhaom2)
Ruiyi Wang (ruiyiw2)

Jiaqi Cheng (jiaqic3)
Jiahui Zhao (jzhao71)

***Abstract*—Voice recognition has gained enormous attention and is applied in various industries. Moreover, Human voice is a natural conduit for emotion. This project aims to recognize the emotion information from human voice by utilizing deep learning algorithms.**

Keywords—voice recognition, emotion, deep learning, 1D-CNN-LSTM, neural networks

I. INTRODUCTION

Voice recognition technologies play an important role in Artificial Intelligence Industry. Most of the widely used AI assistants, such as Alexa, Siri and Cortana, utilize voice recognition technologies to provide new ways for people to interact with the world. As human voice is a natural conduit for emotion, the emotion information gleaned from the tone of voice is widely used to improve the responsiveness of AI assistants during their interactions with humans. Other than the Artificial Intelligence industry, voice emotion recognition also has major application in business practices, medical treatments, and psychological researches. For example, by accurately detecting the emotions from human's voice, the therapist can monitor the mental health status of a patient for PTSD, depression, and suicide signs.

Other than the practical applications and the enormous benefits that voice emotion recognition would bring to the society, by challenging this topic, we will acquire a more thorough understanding of deep learning and neural networks.

The aim of this project is to classify some basic human emotions such as fear, happiness, sadness, surprise and anger, on the dataset '*Toronto emotional speech set'*. Our group will apply knowledge and skills from the STAT 430 class by implementing deep learning models to extract emotion signals and construct feature extraction from audios included in the dataset.

II. RELATED WORK

The challenge in this project is to develop a deep learning model to classify emotions expressed in audio records of two actresses aged 26 and 64 years old. The traditional emotion theory [1], which is pioneered by Tomkins [1962, 1963], believes there exist a small number of basic emotions that determines the emotional expression and responses for all people in the world. The emotion theory considers human emotion as discrete subject, allowing for classification of emotion into six or "seven discrete, basic emotion categories, such as neutral, happiness, sadness, surprise, hear, anger and disgust".

In order to better understand emotional signals within audios, researchers in the fields like computer science, communication engineering and psychology, have worked to combine various kinds of machine learning, especially deep learning algorithms into the speech emotion recognition process in recent years. In their paper [2], author Zhengwei Huang with his colleagues used semi-supervised Convolutional Neural Network, "in which simple features are learned in the lower layers (by contractive convolutional neural network with reconstruction penalization), and affect-salient, discriminative features are obtained in the higher layers" to extract emotional features from audios.

Based on the dataset *Berlin EMO-DB dataset*, an acted speech dataset that is similar to the *Toronto Emotional Speech Set* [3], the authors attempted to extract and examine the "effectiveness of different acoustic features for the recognition of emotions in speech (e.g. spectral, prosodic and voice quality characteristics)" using machine learning models, including k-means unsupervised clustering. In another research paper [4], the author Chenchen Huang and his colleagues explored the new method of feature extraction: extracting features using Deep Belief Network and classifying objects using nonlinear SVM classifier.

However, all these aforementioned research papers are based on the assumption of human emotions as discrete, basic emotion categories. In reality, since speech involved in daily human interaction is complex, subtle and dependent on contexts, some researchers have started exploring a more advanced emotion recognition modelling method, the dimensional analysis [5], which combining the analysis of interactions among "visual (i.e. facial and bodily expression), audio, tactile (i.e. heart rate, skin conductivity, thermal signals etc.) and brain-wave (i.e., brain and scalp signals) modalities". This shift indicates a more integrated, continuous and context-specific interpretation of affective expression in the field of real-world speech emotion recognition.

In this project, we won't go as further as the dimensional approach. We will insist on the categorical approach of emotion recognition and focus on what we have learned from class, the convolutional neural network. However, instead of CNN, we will take into account the continuous characteristics of audio tracks as time series data and implement 1D or 2D CNN model with LSTM. We will refer to the convolutional neural network models used in research paper [6]. But instead of using log-mel spectrogram, we will use mel-frequency cepstral coefficients (MFCCs) to describe the overall shape of a vocal tract, since it may better resemble the resolution of human auditory system compared to the log-mel approach.

III. DATA

The '*Toronto emotional speech set*' dataset (audios are stored in WAV format), includes 2,800 audio files from only female speakers. The dataset was published in 2010 by the psychology department of University of Toronto Mississauga. The recordings are made by two actresses with different ages who made seven different emotions. The whole dataset uses one carrier phrase "Say the word____" with 200 target words.

A. Data Preprocessing

The "*Toronto emotional speech set*" is not in the type of dataset, so it does not have observations or features. There are seven different emotions and two actresses. Each of those emotions has 200 audios per actress. Therefore, we preprocess those audios to extract features from it in order to make them could be fed into any model. The set is based on two female actors with different ages, but we decide to not consider age as a factor, so we mix those audios together. We create a dataset to record the emotion of each audio present and the link toward the audio.

While loading the dataset, the speakers and the emotions are organized in separate folders based on actresses' ages through using the "os" package. For data exploration, we use the "librosa" package to draw the wave plot as the graphical representation of a sound wave vibration in order to see whether significant differences exist among different emotions of the same word from the same actress. Hopefully, without a lot of background noise, there are obvious differences and the dataset is of good quality. Based on our research, we learned that the 'Mel-Frequency Cepstral Coefficient (MFCC)' is a "representation" of the vocal tract that produces the sound. We visualize the MFCC of all emotions of the younger actress on the same word as the wave plots. (Fig. 2.-15.) We found out that those emotions' MFCC graphs are significantly different from each other, and more obvious than the wave plots. It means that MFCC should be an appropriate feature for those audios and can be used to build our model. Therefore, we need to extract MFCC as the feature of audios using "librosa" package. Since we will use two different CNN models (this decision will be further explained in the Methods Section), we process the data differently in order to feed into the models.

- 1D-CNN LSTM:

  For the data used for 1D-CNN LSTM, we extracted the mean of MFCC bands as feature values and cleaned the missing values. The resulting dataset represents 216 features of one audio record, with 2,800 records in total. We then fill the N/A values in the dataset with zeros. The result dataset has 2,800 observations with 217 columns with the corresponding labels and paths.

- 2D-CNN LSTM:

  Compared with 1D-CNN LSTM, there is no need for us to extract the mean of the 20 MFCC bands, so we use the entire MFCC output. Each of the MFCC outputs which is a 2D array (20 bands * 216 audio lengths, which could be viewed as an image with 20* 216 pixels.) Thus, the final dataset of X has a dimension of (2,100, 20, 216).

We then split the data frame into the training dataset and the testing dataset according to an 8:2 ratio. Then, we normalize the training and testing datasets to improve the accuracy and speed up the training process. We also one-hot encoded the labels to convert the categorical data into numeric metrics for modeling. Following that, we convert all train and test dataset to NumPy arrays for modeling. At last, we expand those data frame by one dimension to present the channel, which means that the dimensions of the two input training data frame (the X) are (2,100, 215, 1) and (2,100, 20, 216, 1).

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	-0.044627	0.503193	1.832104	1.033470	-0.087090	-0.675430	-0.482462	-0.139783	-0.639739	-0.763807	-0.581303	-0.558931	-0.679260	-0.365469	-0.212925	0.252215	0.444523	0.233881
0	1.408385	1.394968	-0.202459	-0.195073	-0.370148	-0.794728	-1.294341	-0.922862	-0.484884	-0.105733	0.085857	-0.088750	-0.414729	0.054110	-0.187387	-0.015469	-0.143869	-0.285807
0	-1.280938	-1.722273	-1.645818	-0.984249	-0.271779	0.264881	0.221787	0.218768	0.441080	0.304798	0.118488	-0.063079	-0.200261	0.413393	0.770818	0.900423	0.768884	0.897308

3 rows × 215 columns

Fig. 1. First three obeservations of the preprocessed dataset for 1D-CNN LSTM

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	0.085561	0.018192	-0.047549	-0.094307	-0.131591	-0.162259	-0.190917	-0.219205	-0.249731	-0.279915	-0.299984	-0.319528	-0.339276	-0.358262	-0.381597	-0.404838	-0.426514	-0.450192
1	-0.157467	-0.204863	-0.249318	-0.279800	-0.307387	-0.331361	-0.354858	-0.377890	-0.403412	-0.426809	-0.445218	-0.463739	-0.483906	-0.501970	-0.523593	-0.543412	-0.563964	-0.587815
2	0.035532	0.084024	0.112452	0.153078	0.191701	0.222386	0.238418	0.259167	0.277893	0.298717	0.309455	0.318489	0.329794	0.335310	0.321803	0.322309	0.329289	0.334874
3	-0.128005	-0.169698	-0.216728	-0.248601	-0.275952	-0.295068	-0.313177	-0.339243	-0.363274	-0.385002	-0.404370	-0.419148	-0.438456	-0.455069	-0.473932	-0.496074	-0.515434	-0.531938
4	-0.028633	-0.038689	-0.040990	-0.055913	-0.082322	-0.087881	-0.104229	-0.118271	-0.116715	-0.114799	-0.115477	-0.110816	-0.111271	-0.125090	-0.129771	-0.128620	-0.130799	-0.125420
5	0.006740	0.021074	-0.005573	-0.031383	-0.042165	-0.061617	-0.088921	-0.084656	-0.084926	-0.103149	-0.117735	-0.126828	-0.134908	-0.140277	-0.145885	-0.156034	-0.169878	-0.177610
6	0.117249	0.156326	0.195037	0.215822	0.237845	0.252758	0.267551	0.276479	0.290576	0.299007	0.301067	0.295434	0.290448	0.292506	0.294489	0.283817	0.277897	0.275282
7	0.114332	0.148519	0.188431	0.211751	0.233467	0.259951	0.279259	0.298909	0.319504	0.340018	0.354807	0.366030	0.377608	0.390700	0.403798	0.414109	0.427170	0.441705
8	0.023980	0.020063	0.000177	-0.023558	-0.039926	-0.054874	-0.062735	-0.068624	-0.091730	-0.092149	-0.106536	-0.116728	-0.108880	-0.097514	-0.096075	-0.099511	-0.097040	-0.087284
9	0.127460	0.165963	0.209088	0.233624	0.261562	0.280146	0.302087	0.326469	0.350871	0.377244	0.391790	0.403133	0.420054	0.434861	0.448642	0.460982	0.477759	0.496308
10	0.085003	0.111373	0.146289	0.167180	0.195377	0.209163	0.219965	0.239935	0.261587	0.279078	0.289556	0.292986	0.303864	0.311379	0.320857	0.326663	0.335566	0.360373
11	0.085780	0.098674	0.122066	0.148897	0.184080	0.203183	0.223683	0.243984	0.264166	0.288067	0.305064	0.305805	0.315188	0.330174	0.351582	0.371352	0.379339	0.383532
12	0.072617	0.098394	0.131333	0.176298	0.199091	0.214784	0.237608	0.248757	0.272940	0.296092	0.318569	0.320340	0.336798	0.363044	0.394847	0.421550	0.429331	0.439975
13	-0.017909	-0.014978	-0.021004	-0.000090	-0.014181	-0.028176	-0.029835	-0.030468	-0.027751	-0.007624	0.010327	0.015896	0.014105	0.035739	0.061428	0.082112	0.086589	0.093796
14	0.046338	0.059056	0.093135	0.137404	0.141535	0.158820	0.169612	0.185949	0.182745	0.211809	0.239548	0.265383	0.278810	0.292368	0.300349	0.325686	0.347877	0.365403
15	0.016271	0.022813	0.040476	0.042287	0.047552	0.033420	0.043101	0.055968	0.042949	0.051507	0.074443	0.081816	0.081318	0.091560	0.073588	0.072609	0.095363	0.131073
16	-0.014941	-0.002782	0.011947	0.029657	0.041561	0.040969	0.063352	0.087314	0.075050	0.070733	0.089177	0.105536	0.107242	0.110159	0.103638	0.109920	0.125336	0.144599
17	-0.003090	0.001100	0.025159	0.053750	0.054334	0.053776	0.065078	0.085419	0.071584	0.056555	0.085253	0.079327	0.073684	0.075420	0.068745	0.087720	0.080833	0.106183
18	0.068248	0.123099	0.133482	0.147839	0.155062	0.146879	0.149427	0.178208	0.191718	0.210461	0.220238	0.228989	0.220942	0.219844	0.225585	0.226720	0.228841	0.248584
19	-0.001246	0.052075	0.050463	0.055114	0.082162	0.097102	0.087615	0.090201	0.118489	0.144013	0.146295	0.146943	0.138076	0.122245	0.133746	0.136930	0.127598	0.123223

20 rows × 216 columns

Fig. 2. First MFCC output of the preprocessed dataset for 2D-CNN LSTM

Therefore, compared with the 1D-CNN LSTM, more features could be included in the 2D-CNN LSTM model. We assumed that the more features included will help improve the accuracy rate of 2D-CNN LSTM compared to ID-CNN LSTM.

B. Wave Plot and Data Visualization of MFCC

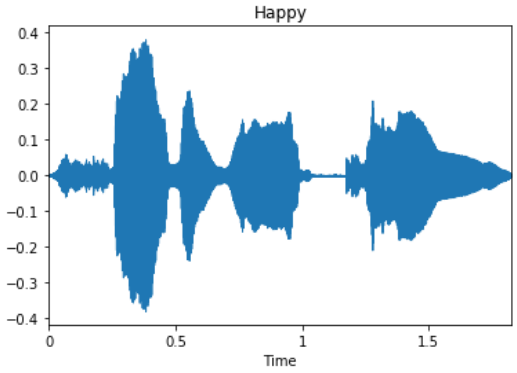

Fig. 3. Wave plot: younger actress say the word "king" with emotion Happy

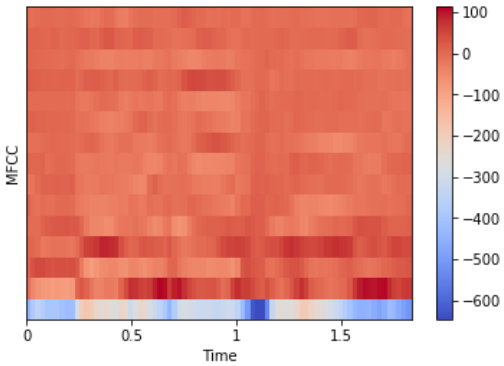

Fig. 4. Data Visualization of MFCC: Happy

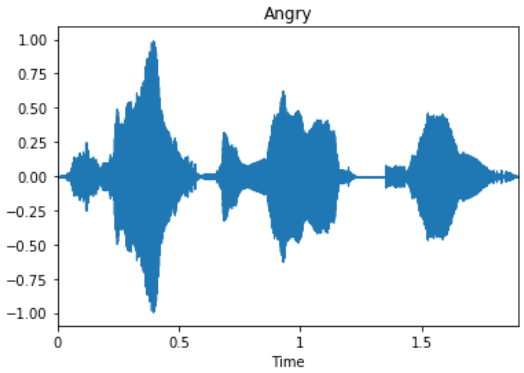

Fig. 5. Wave plot: younger actress say the word "king" with emotion Angry

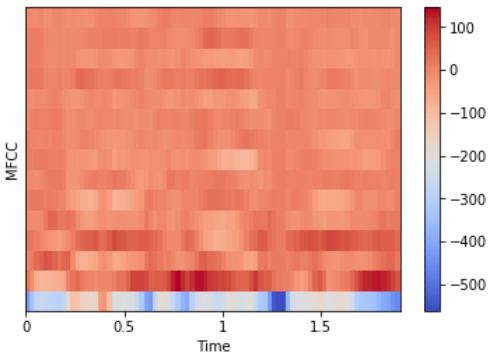

Fig. 6. Data Visualization of MFCC: Angry

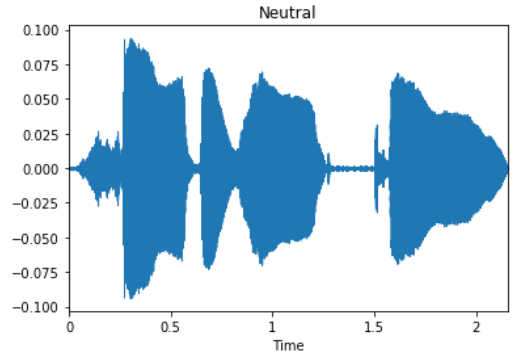

Fig. 7. Wave plot: younger actress say the word "king" with emotion Neutral

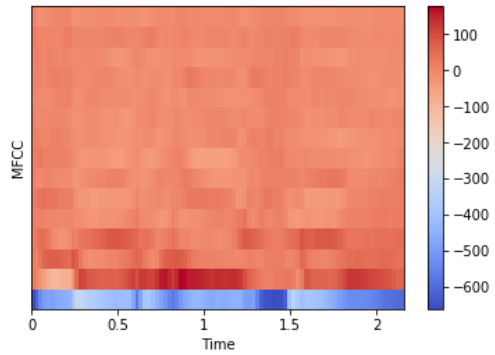

Fig. 8. Data Visualization of MFCC: Neutral

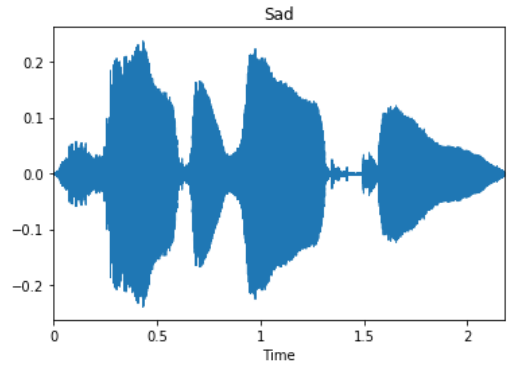

Fig. 9. Wave plot: younger actress say the word "king" with emotion Sad

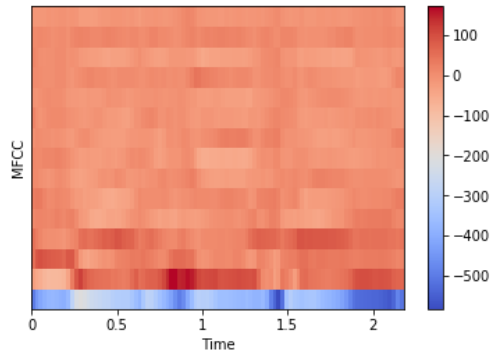

Fig. 10. Data Visualization of MFCC: Sad

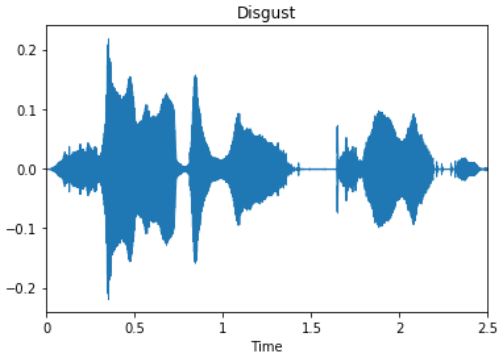

Fig. 11. Wave plot: younger actress say the word "king" with emotion Disgust

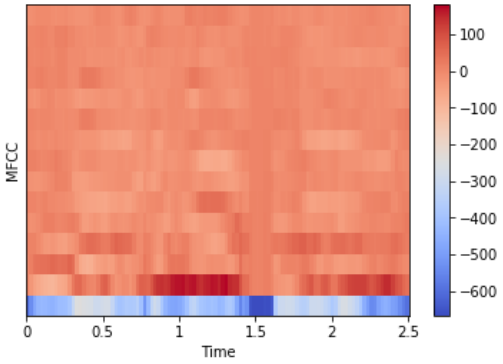

Fig. 12. Data Visualization of MFCC: Disgust

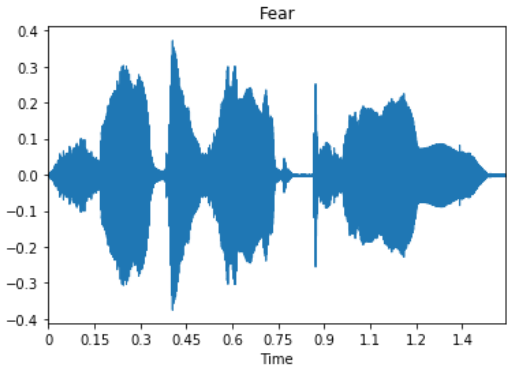

Fig. 13. Wave plot: younger actress say the word "king" with emotion Fear

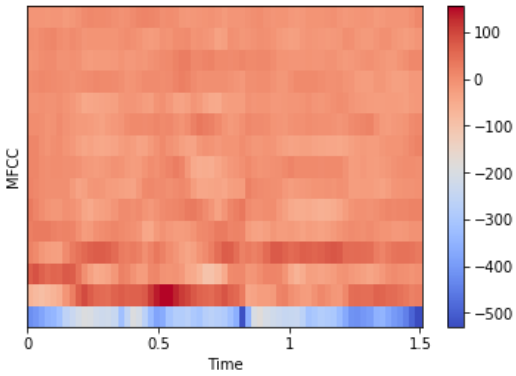

Fig. 14. Data Visualization of MFCC: Fear

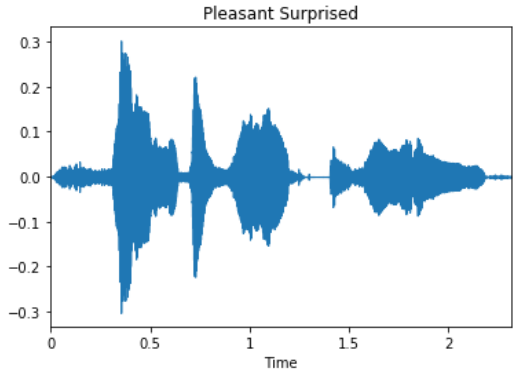

Fig. 15. Wave plot: younger actress say the word "king" with emotion Pleasant Surprised

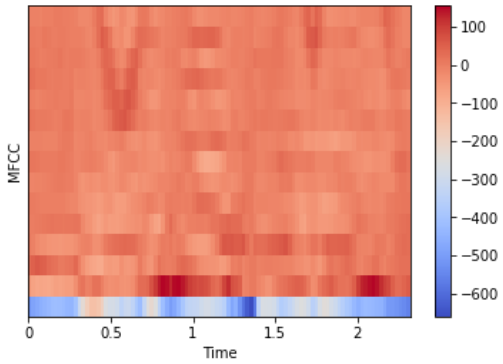

Fig. 16. Data Visualization of MFCC: Pleasant Surprised

IV. Methods

A. *Proposed Network Architectures*

The convolutional neural networks work well for identifying simple patterns within the data that are used to further develop more complex patterns in deeper layers. Therefore, the CNN method will play an important role in this project. Based on the prior research, a 1D-CNN is very effective with Natural Language Processing, as the audio signals are set in fixed-length periods. On the other hand, the 2D-CNN brings a larger storage of information in the MFCC bands.

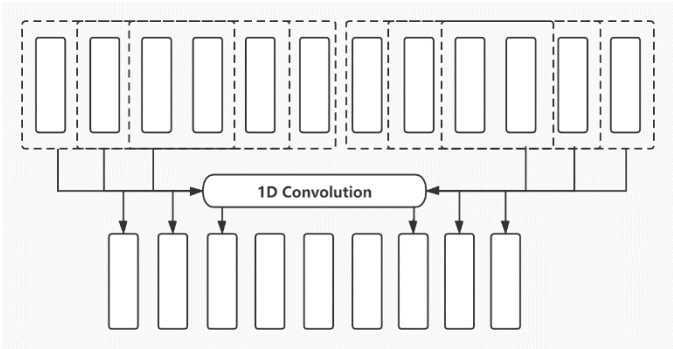

Fig. 17. Diagram of 1D Convolution with kernel size of 4 and stride 1

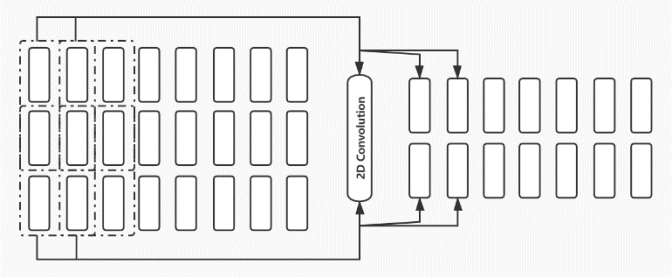

Fig. 18. Diagram of 2D Convolution with kernel size 2x2 and stride 1x1

Moreover, the Long-term Short-term memory method (LSTM) is specialized for processing a sequence of values, which each member of the learned feature is a function of the previous members of the output. The combination of the CNN and LSTM can learn the high-level features, which contain both the local information and the long-term contextual dependencies. These differences between the 1D & 2D-CNN and the uniqueness of LSTM derived our determination to build the following models: 1D-CNN LSTM and 2D-CNN LSTM.

Although we use two networks, they indeed have the similar architectures, both consisting of four parts, two convolutional layers, one LSTM layer and one fully connected layer. In addition, in order to prevent overfitting for the 2D-CNN LSTM model, we use dropout nodes for our 2D-CNN LSTM model.

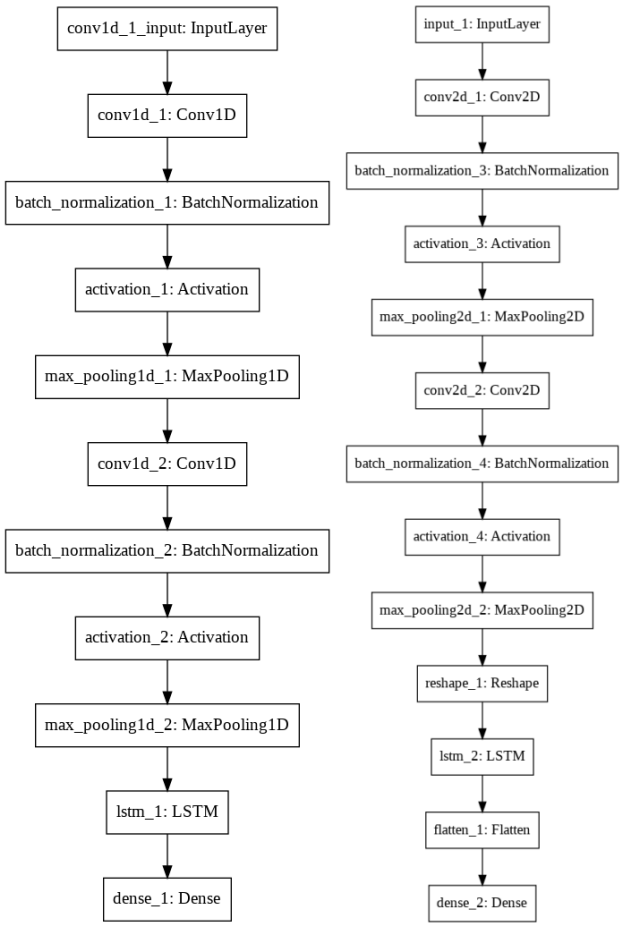

Fig. 19. Proposed Network Architectures

Each 1D/2D-CNN LSTM layer contains four parts: one convolution, one ELU activation, one Batch Normalization, and one Max Pooling.

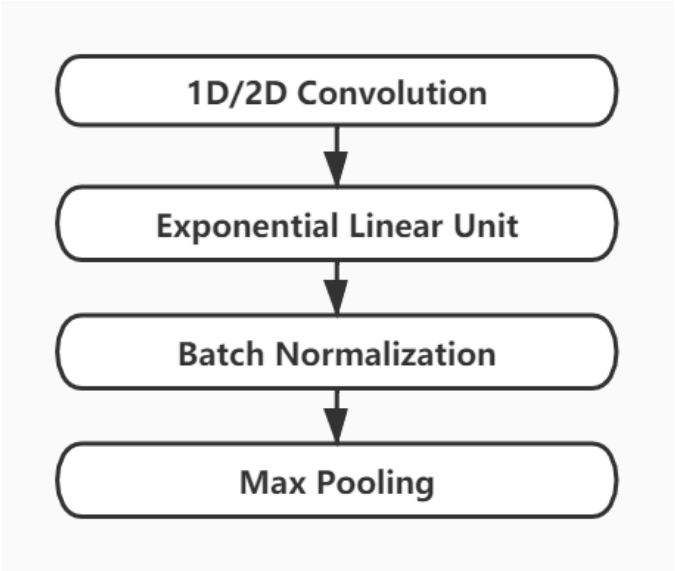

Fig. 20. Composition of the 1D/2D-CNN Blocks

B. CNN and LSTM

The input of 2D convolution layer is x(i,j), the convolution kernel w(i,j) will be randomly initialized. Then calculate the convolved feature

$$z(i,j) = x(i,j) * w(i,j),$$

and put it into ELU, which is

$$\begin{cases} x, & if\ x > 0 \\ \alpha(exp(x) - 1), & if\ x \leq 0 \end{cases}$$

where alpha is a positive constant.

Then, we normalize the training and testing datasets to improve the accuracy and speed up the training process by the technique of batch normalization, which is

$$z_i^l = \sigma(BN(b_i^l + \sum_j z_i^{l-1} * w_{ij}^l))$$

At last, the features will put into the max pooling. We plan to use kernel size of 3x3 and stride size 1x1 for both convolution layers. The first convolution layer has 64 filters and the second has 32 filters. For the max pooling layer, we use pool size 4x4 and stride 4x4 for the first layer, and pool size 2x2 and stride size 2x2 for the second layer.

The output features will be flattened and then go through LSTM structure. The LSTM is presented by functions as below:

$$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f)$$
$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i)$$
$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o)$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \sigma_t(W_c x_t + U_c h_{t-1} + b_c)$$

$$h_t = o_t \circ \sigma_t(c_t)$$

where x_t is the input vectors, f_t, i_t, o_t are the forget, input, and output gate vectors, c_t is the cell, and h_t is the output.

Activation functions sigmoid and hyperbolic tangent are used. The output from the LSTM will be put into the fully connected layer with a SoftMax function, which is

$$S_j(z) = \frac{e^{j_z}}{\sum_{i=1}^{n(l)} e^{z_i}}$$

The SoftMax function output then will be used to analyze the performance (accuracy rate) of our neural network. The dimensions of each layer in both networks are displayed below:

Layer (type)	Output Shape	Param #
conv1d_1 (Conv1D)	(None, 215, 64)	256
batch_normalization_1 (Batch)	(None, 215, 64)	256
activation_1 (Activation)	(None, 215, 64)	0
max_pooling1d_1 (MaxPooling1)	(None, 53, 64)	0
conv1d_2 (Conv1D)	(None, 53, 32)	6176
batch_normalization_2 (Batch)	(None, 53, 32)	128
activation_2 (Activation)	(None, 53, 32)	0
max_pooling1d_2 (MaxPooling1)	(None, 13, 32)	0
lstm_1 (LSTM)	(None, 64)	24832
dense_1 (Dense)	(None, 7)	455

Fig. 21. Dimensions of each layer in the 1D-CNN LSTM model

Layer (type)	Output Shape	Param #
input_2 (InputLayer)	(None, 20, 216, 1)	0
conv2d_3 (Conv2D)	(None, 20, 216, 64)	640
batch_normalization_11 (Batch)	(None, 20, 216, 64)	256
activation_11(Activation)	(None, 20, 216, 64)	0
max_pooling2d_3 (MaxPooling2)	(None, 10, 108, 64)	0
conv2d_4 (Conv2D)	(None, 10, 108, 32)	18464
batch_normalization_12 (Batch)	(None, 10, 108, 32)	128
activation_12 (Activation)	(None, 10, 108, 32)	0
max_pooling2d_4 (MaxPooling2)	(None, 5, 54, 32)	0
reshape_2 (Reshape)	(None, 270, 32)	0
lstm_1 (LSTM)	(None, 270, 64)	24832
flatten_2 (Flatten)	(None, 17280)	0
dense_6 (Dense)	(None, 7)	120967

Fig. 22. Dimensions of each layer in the 2D-CNN LSTM model

RESULTS/DISCUSSION

Since we are doing classification of speech emotion, the summary table below shows the accuracy rates of all models.

Classification Report (1D)

	precision	recall	f1-score	support
angry	0.00	0.00	0.00	98
disgust	0.46	0.89	0.60	94
fear	0.45	0.87	0.59	94
happy	0.20	0.08	0.12	106
neutral	0.33	0.75	0.46	106
sad	0.70	0.13	0.23	104
surprise	0.10	0.03	0.05	98
accuracy			0.39	700
macro avg	0.32	0.39	0.29	700
weighted avg	0.32	0.39	0.29	700

Fig. 23. Summary table of the 1D-CNN LSTM model

Classification Report (2D)

	precision	recall	f1-score	support
angry	1.00	0.92	0.96	98
disgust	1.00	0.98	0.99	94
fear	0.91	0.99	0.95	94
happy	0.95	0.98	0.96	106
neutral	0.95	0.99	0.97	106
sad	0.99	0.99	0.99	104
surprise	0.98	0.92	0.95	98
accuracy			0.97	700
macro avg	0.97	0.97	0.97	700
weighted avg	0.97	0.97	0.97	700

Fig. 24. Summary table of the 2D-CNN LSTM model

Classification Report (2D with Regularization))

	precision	recall	f1-score	support
angry	0.89	0.87	0.88	98
disgust	0.97	0.96	0.96	94
fear	0.70	1.00	0.82	94
happy	0.97	0.74	0.84	106
neutral	0.93	0.95	0.94	106
sad	0.94	0.98	0.96	104
surprise	0.94	0.77	0.84	98
accuracy			0.89	700
macro avg	0.91	0.89	0.89	700
weighted avg	0.91	0.89	0.89	700

Fig. 25. Summary table of the 2D-CNN LSTM model with Regularization

```
for 1D CNN (with LSTM) Model
[[ 0  7 53  3 34  0  1]
 [ 0 84  0  0  7  3  0]
 [ 0  0 82  6  4  0  2]
 [ 0  1 23  9 63  0 10]
 [ 0  6  8  5 79  3  5]
 [ 0 71  4  0  7 14  8]
 [ 0 15 13 21 46  0  3]]
```

Fig. 26. Confusion matrix of the 1D-CNN LSTM model

```
Confusion Matrix for 2D CNN (with LSTM) Model
[[ 90   0   7   0   1   0   0]
 [  0  92   0   0   1   0   1]
 [  0   0  93   1   0   0   0]
 [  0   0   1 104   0   0   1]
 [  0   0   0   0 105   1   0]
 [  0   0   0   0   1 103   0]
 [  0   0   1   5   2   0  90]]
```

Fig. 27. Confusion matrix of the 2D-CNN LSTM model

```
Confusion Matrix for 2D CNN (with LSTM and Regularization) Model
[[ 85   0  13   0   0   0   0]
 [  1  90   0   1   1   1   0]
 [  0   0  94   0   0   0   0]
 [  3   0  18  78   2   0   5]
 [  0   0   0   0 101   5   0]
 [  0   0   0   0   2 102   0]
 [  7   3   9   1   3   0  75]]
```

Fig. 28. Confusion matrix of the 2D-CNN LSTM model with Regularization

From the summary table above we can find out that the 1D-CNN LSTM network performs much worse than we expected. As we extract the mean of the 20 MFCC bands to feed into the 1D-CNN LSTM model, many features are not included into the 1D-CNN LSTM model.

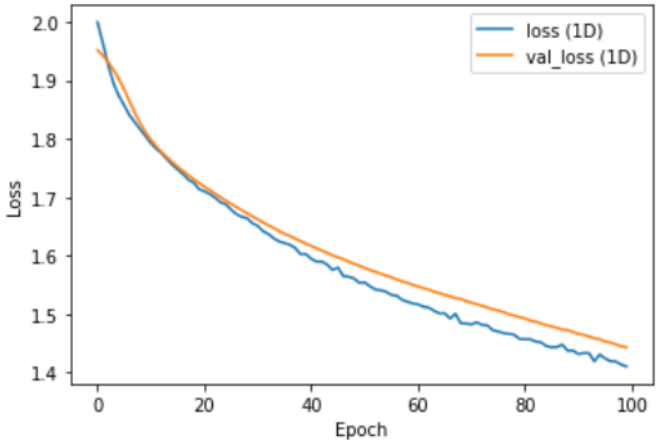

Fig. 29. Loss function graph of the 1D-CNN LSTM model

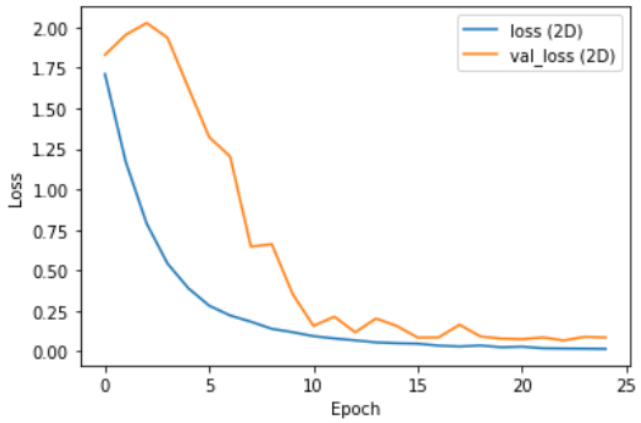

Fig. 30. Loss function graph of the 2D-CNN LSTM model

Based on the training history plot above, the recognition accuracy for 1D-CNN LSTM model has an increasing trend. To be more specific, the categorical accuracy for the validation data shows to increase from 15.48% to 40.00%, and the categorical accuracy for the training data rises sharply from 9.58% to 42.38%. By contrast with the result of 1D-CNN LSTM network, the 2D-CNN LSTM network achieves recognition accuracies of 97.62% and 99.88% on the validation data and training data respectively, which compare favorably to the accuracy of 40.00% and 42.38% obtained by the 1D-CNN LSTM approaches. The effects of adding the Dropout nodes to the 2D-CNN LSTM can be viewed from the training history as well. Via using the Dropout nodes, the accuracy rate has decreased around 10%. The validation accuracy rate dropped to 89.76% and the training accuracy rate dropped to 88.57%, indicating that the overfitting issue has been fixed to some extent.

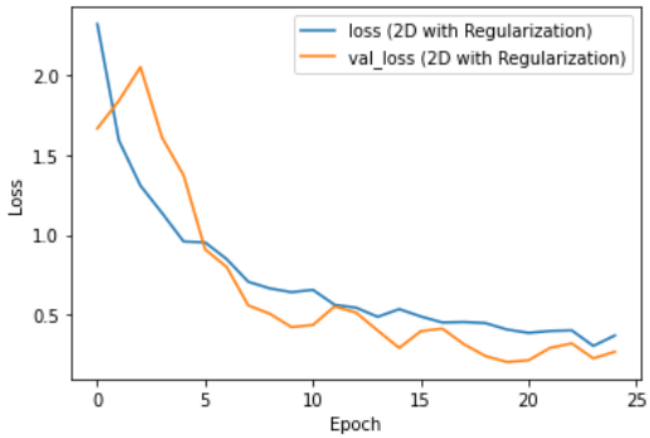

Fig. 31. Loss function graph of the 2D-CNN LSTM model with Regularization

Conclusion/Peer Evaluation

Based on the results we get from this analysis; it is obvious that the 2D-CNN LSTM network outperforms the 1D-CNN LSTM network. It is understandable that 1D-CNN LSTM model is time efficient as it only uses the mean of MFCC bands. On the contrary, the whole MFCC outputs of each audio, which is a 2D matrix, are being used for the 2D-CNN LSTM model. Thus, it greatly raises the accuracy rate but also consumes more time and introduced the problem of overfitting. In a nutshell, the 2D-CNN LSTM model with Regularization of Dropout nodes ensures a high accuracy rate while avoiding the issue with overfitting.

All our group members have learned a lot from this project, including dealing with audios, extract MFCC features from audios and treat MFCC output as an image to do further analysis. Although we are very optimistic and confident with our outcomes, problems like overfitting might still exist during our analysis.

We have used the Dropout Regularization to prevent overfitting in this analysis. However, we believe that there may exist better methods which could be more efficient to deal with the issue of overfitting. In the future, in order to make our 2D-CNN LSTM network more reliable, it is necessary to use more

dataset that contains people with different genders, ages, and accents to train the model for further improvement and analysis.

PEER EVALUATION

All group members contributed equally in this project. We all decide to assign each group member 25% of contribution to this stage of the project.

Jiahui Zhao: Background research, model training;

Victoria Meng: Data Preprocessing, data visualization;

Ruiyi Wang: Model building; model architecture construction;

Jiaqi Cheng: Results compiling, model architecture construction;

REFERENCES

[1] Keltner, D., & Ekman, P. (2000). Facial Expression of Emotion. In M. Lewis, & J. Haviland-Jones (Eds.), Handbook of Emotions (2nd ed., pp. 236-249). New York: Guilford Publications, Inc.

[2] Zhengwei Huang, Ming Dong, Qirong Mao & Yongzhao Zhan, "Speech Emotion Recognition Using CNN", MM '14: Proceedings of the 22nd ACM international conference on Multimedia, Nov. 2014, pp. 801-804 https://doi.org/10.1145/2647868.2654984

[3] I. Luengo, E. Navas & I. Hernáez, "Feature Analysis and Evaluation for Automatic Emotion Identification in Speech", in IEEE Transactions on Multimedia, vol. 12, no. 6, pp. 490-501, Oct. 2010.

[4] Chenchen Huang, Wei Gong, Wenlong Fu & Dongyu Feng, "A Research of Speech Emotion Recognition Based on Deep Belief Network and SVM", Mathematical Problems in Engineering, vol. 2014, 12 Aug 2014 https://doi.org/10.1155/2014/749604

[5] Gunes, Hatice & Pantic, Maja. (2010). Automatic, Dimensional and Continuous Emotion Recognition. IJSE. 1. 68-99. 10.4018/jse.2010101605.

[6] Jianfeng Zhao, Xia Mao & Lijiang Chen. (2019) Speech Emotion Recognition Using Deep 1D & 2D CNN LSTM Networks, vol. 47, Jan. 2019, pp. 312-323

[7] Kate Dupuis & M. Kathleen Pichora-Fuller, Toronto Emotional Speech Set (TESS), University of Toronto, Psychology Department, 2010, https://tspace.library.utoronto.ca/handle/1807/24487