Speech Emotion Recognition Using LSTM Neural Network

1st Hesham Elshafey
Electrical and Computer Engineering
University of Rochester
helshafe@u.rochester.edu

***Abstract*—This project implements a machine learning model using Long Short-term Memory Recurrent Neural Networks to detect and classify emotional state from speech. The emotional classes includes neutral, calm, happy, sad, angry, fearful, disgust, and surprised, with two intensity classes of normal and intense. The model was trained on the RVADESS [1] dataset audio-only portion. The audio files were augmented by creating variants of each file that includes noise, shifted pitch, and time stretching to diversify the dataset and enhance the model training. The final model architecture reached a validation accuracy of 73% on 125 epoches, early stopping patience of 10, and 256 units in the first LSTM layer.**

***Index Terms*—LSTM, Emotions, Data Augmentation**

I. Introduction

Humans ability to infer emotions from speech, even over a phone call with heavy audio processing happening on the real voice, is astronomical. Emulating this capability in machines through computational models is a growing area of research within artificial intelligence and machine learning. This project focuses on implementing a machine learning model using Long Short-Term Memory (LSTM) Recurrent Neural Networks (RNNs) to detect and classify emotional states from speech. Speech Emotion Recognition (SER) has applications in various domains, such as enhancing user experience in virtual assistants, improving mental health monitoring systems, and advancing human-computer interaction. The emotional classes considered in this project include neutral, calm, happy, sad, angry, fearful, disgust, and surprised, each with two intensity levels: normal and intense. The dataset used for training and evaluation is the audio-only portion of the Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) [1]. The RAVDESS dataset is well-known for its high-quality recordings of emotional expressions by professional actors, making it a valuable resource for SER tasks. The extract audio feature used to train the model is the Mel-frequency Cepstral Coefficients (MFCC). MFCC features capture the frequency and magnitude content of audio in a way that mimics human perception of audio. These features captures the sudden energy bursts produced by an angry person while also capturing the smooth frequency and magnitude transitions of a calm person. To improve the generalization capability of the model, the dataset was augmented by generating variants of each audio file. These variants included added noise, pitch shifting, and time stretching, which introduce variability in the data and simulate real-world conditions. Data augmentation is a critical step in mitigating overfitting and improving the robustness of the model. The LSTM RNN architecture was chosen due to its strength in capturing temporal dependencies in sequential data, such as speech signals. The final model architecture incorporated a robust configuration with 256 units in the first LSTM layer, achieving a validation accuracy of 73%. The model was trained for a maximum of 125 epochs, with early stopping applied to halt training if validation performance did not improve after 10 consecutive epochs.

II. Method

A. The dataset: RAVDESS

The data set consists of 24 sub-folders, each includes voice recordings from a specific actor. Each actor have multiple recording saying either ”Kids are talking by the door” or ”Dogs are sitting by the door”. Each recording from a given actor exhibits one of the 8 emotional classes mentioned above and one of the two emotional intensity classes.Plots of the emotional classes distribution and intensity distribution are shown below in Figures 1 and 2 respectively. The gender distribution is symmetrical. These features of the dataset including symmetry and professional acting makes it very robust to train the speech emotion recognition model. The dataset also doesn’t include an explicit CSV file for labels. The labels are encoded in the file name itself, with decoding table provided by the dataset publisher. An example of label decoding from a file is shown below in Figure 3.

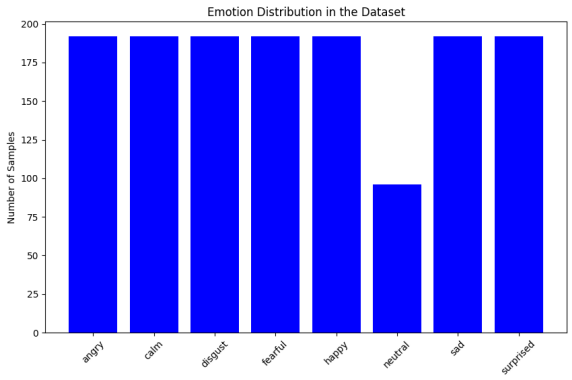

Fig. 1. Emotional Distribution in the Dataset

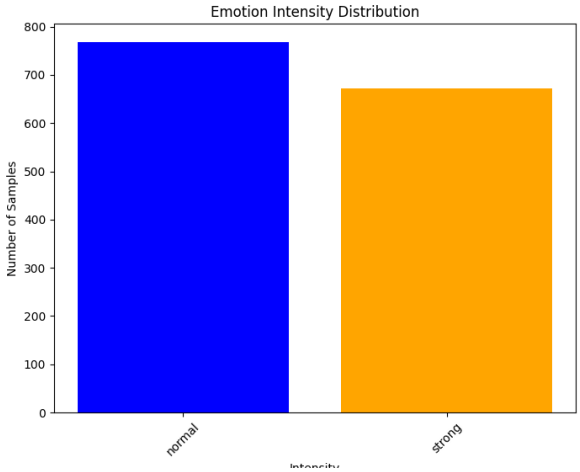

Fig. 2. Emotional Intensity Distribution in the Dataset

Filename example: 03-01-06-01-02-01-12.wav

1. Audio-only (03)
2. Speech (01)
3. Fearful (06)
4. Normal intensity (01)
5. Statement "dogs" (02)
6. 1st Repetition (01)
7. 12th Actor (12)

Female, as the actor ID number is even.

Fig. 3. Example of Labels Decoding From File Name

B. Mel-Frequency Cepstral Coefficients (MFCC)

As mentioned above, MFCC features offer great insight into emotional status of the speaker by capturing the transitions in frequency and intensity over time. This transitions are the perfect input for an LSTM-based model that depends on sequential, temporal data. Emotions like anger and surprise are linked to discontinuous changes in the frequency with higher energy bursts apparent in the peaks of higher order MFCC coefficients. On the other hand, emotions like sadness and calmness exhibits smoother frequency transitions with energy mostly in the lower frequency domain, manifesting in the first few MFCC coefficients. Figures 4, 5, and 6 represents three MFCC plots generated from the data set for the same actor being calm, angry, and very angry (angry with the intensity class ”intense”). The calm plot shows much smoother frequency transitions depicted by the lighter red and orang bins. The blue pins also shows a smooth transition in the frequency ranges. In addition, the overall energy and voice magnitude is low. On the other hand, the very angry plot exhibits very sharp frequency and magnitude transitions manifesting in the dark red and orang bins. The very angry plot also exhibits higher order MFCC coefficients (about 11 bins) compared with the calm plot (about 5 bins). Finally, the very angry plot shows sharp intensity peaks compared with the calm plot. These insights offered by the MFCC features can help train a decent model for speech emotion recognition; however, MFCC features alone may not but robust against opposite emotions that look alike. For example, calm and sad emotions have roughly the same MFCC features despite being two completely opposite emotional states. Given the goal of building a light model, no further features are incorporated in the model.

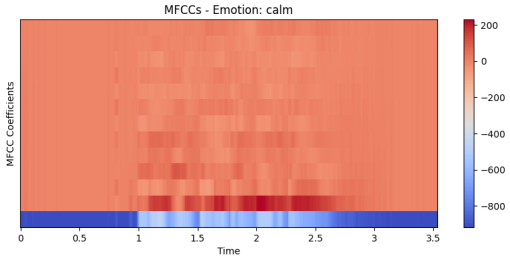

Fig. 4. MFCC plot for a calm actor

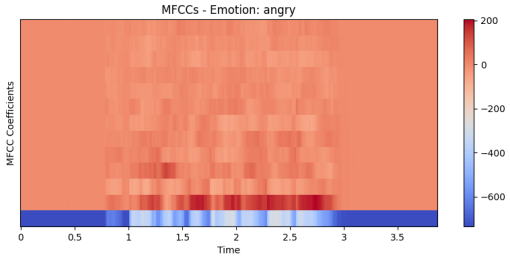

Fig. 5. MFCC plot for an angry actor

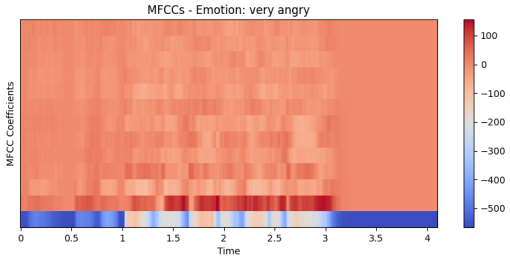

Fig. 6. MFCC plot for a very angry actor

C. Data Preparation

In the data-processing.py file, a data preparation function was used to iteratively go through actor folder and load each audio file. In the initial model, the MFCC extraction function was immediately called on the loaded audio file, but as will be presented later, the final model performed data augmentation on the loaded file to create three variants of the file: noisy,

pitch shifted, and time stretched versions. This function also performed label extraction using the file name encoding by reading the file name with a ”-” delimiter, essentially splitting the file name into its atomic arguments. The function finally returns the extracted MFCC features for all the files along with their labels. A rough flow chart of the data preparation step is shown in Figure 7 below.

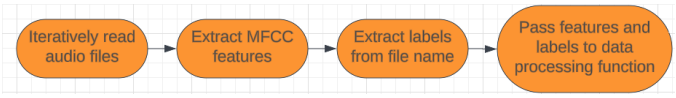

Fig. 7. Data Preparation Steps

D. Data Preparation

In the data preparation function, a LabelEncoder() object was created to transform the labels for LSTM input. The dataset was splitted into train and test sets with 20% of the dataset for testing. As I learned in my presentation, while that was accidental, splitting the data with the actor folder name (no overlapping between train and test) was an efficient decision because the testing would be on audio files the model have never seen before, hence testing the model generality. A chart of the data processing step is shown below in Figure 8.

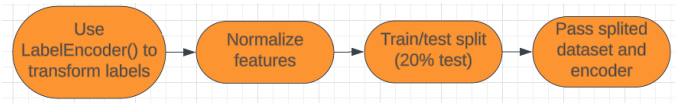

Fig. 8. Data Processing Steps

E. Training and Evaluation

For the training step, the input was reshaped to appropriately match the LSTM layers input shape. The model was then created and printed for evaluation purposes. Finally the model was trained and model history including F-1 score, accuracy, and validation was collected. For the evaluation, plots for training and test accuracy and loss are printed to evaluate the model and add enhancements if needed. Flow charts for training and evaluation are shown in the figures below.

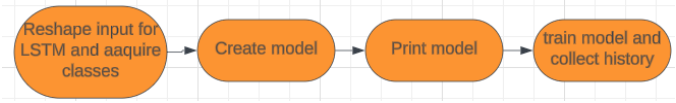

Fig. 9. Model Training

Fig. 10. Model Evaluation

F. General Model Architecture

The general model, beyond data preparing, processing, and MFCC extraction, consists of two LSTM layers, three dropout layers, and two dense layers with RELU and Softmax activations. The decision to use such architecture stems from the main goal of building a simple speech emotion recognition model without over engineering the model layers. Many SER models attach CNN and more LSTM/dropout layers to the models for better accuracy and to avoid over fitting. The results section below will discuss the iterative refining process of the model. A chart of the general model architecture is shown in Figure 11 below.

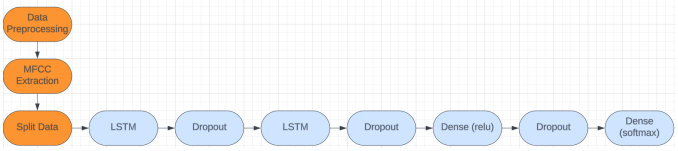

Fig. 11. General Model Architecture

III. Results

The initial model follows the general model layer setup with input LSTM layer of (13, 128) shape followed by a dropout layer of (13, 128) shape. Another LSTM/dropout pair is added next with shape (64) followed by dense/dropout pair of shape (128). Finally, a final dense layer with softmax activation is attached to the model with shape of (8). The initial model architecture is shown in figure 12 below. Figure 13 shows the validation and training accuracy and loss as a function of epoches. As apparent, the validation accuracy is below 35%, which is way below the 70% accuracy goal this project is aiming for. The first refinement step was to add data augmentation to the data preparing step to increase the diversity of the data. AddNoise(), ShiftPitch(), and StretchTime() functions were created to generate a noisy, pitch shifted, and time stretched versions of each file, essentially multiplying the dataset by four and training the model on realistic audios that usually include noise and other features not reflected in the controlled environment where the actors recorded the audio. The updated data preparation step is shown below in Figure 14.

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 13, 128)	66,560
dropout (Dropout)	(None, 13, 128)	0
lstm_1 (LSTM)	(None, 64)	49,408
dropout_1 (Dropout)	(None, 64)	0
dense (Dense)	(None, 128)	8,320
dropout_2 (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 8)	1,032

Total params: 125,320 (489.53 KB)
Trainable params: 125,320 (489.53 KB)
Non-trainable params: 0 (0.00 B)

Fig. 12. Initial Model

Fig. 13. Initial Model Results

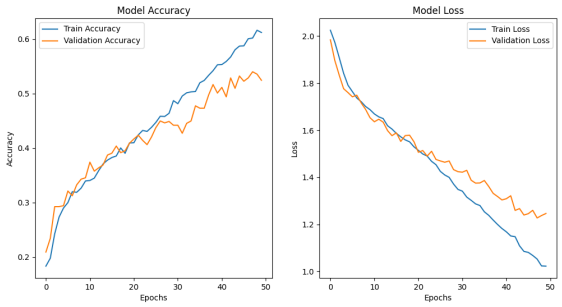

Fig. 15. Initial Model with Data Augmentation

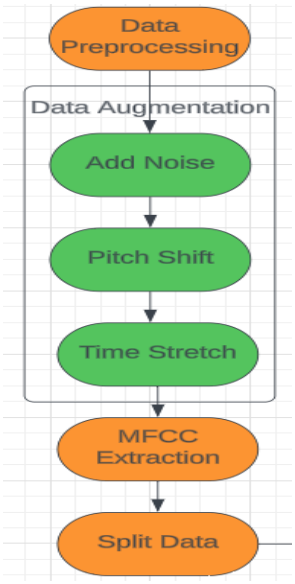

Fig. 14. Data Augmentation

The same model architecture above was used with the new data preparation technique that incorporates data augmentation into the dataset before the training. The model was retrained with 50 epoch and early stopping patience of 5. The evaluation results were collected from the model history and are shown below in Figure 15. The data augmentation step significantly enhanced the model accuracy on validation, reaching approximately 55%; however, this accuracy was still below the desired goal of 70% accuracy. The second refinement step was to alter the model itself and its training parameter in attempt to achieve higher accuracy. The initial LSTM layer units were doubled to be 256 instead of 128. The number of training epoch was increased to 200 while increasing the patience to 10. Increasing the patience allows the model more time to correct its weights in case the validation accuracy starts to degrade, hoping for a weight correction to happen. The new model architecture is shown below in Figure 15. Also this new model required almost double the training time, it yielded very good results as shown in Figure 16 below. The validation accuracy recorded for this model was approximately 73%, surpassing the 70% accuracy goal set for this project.

Layer (type)	Output Shape	Param #
lstm_2 (LSTM)	(None, 13, 256)	264,192
dropout_3 (Dropout)	(None, 13, 256)	0
lstm_3 (LSTM)	(None, 64)	82,176
dropout_4 (Dropout)	(None, 64)	0
dense_2 (Dense)	(None, 128)	8,320
dropout_5 (Dropout)	(None, 128)	0
dense_3 (Dense)	(None, 8)	1,032

Total params: 355,720 (1.36 MB)
Trainable params: 355,720 (1.36 MB)
Non-trainable params: 0 (0.00 B)

Fig. 16. Final Model

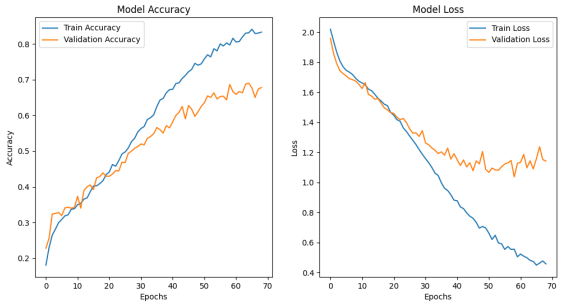

Fig. 17. Final Model Results

The model's full training summary that includes the accuracy percentages for each emotional class, precision, support, F-1 score, and final model accuracy is shown in Figure 18 below. The final full architecture of this project including the model shape and all pre and post processing steps is shown in Figure 19 below. The final model was saved as a .keras file that can be loaded into any other script and used to generate inferences fast. An example script was created to test the model deployment by choosing random 18 files from the dataset and passing them to the loaded .keras model for inference. The generated results of this script is shown below in Figure 20.

```
Test Loss: 1.0247074365615845
Test Accuracy: 0.7265625
36/36 ━━━━━━━━━━━━━━━━━━━━ 1s 9ms/step
Classification Report:
              precision    recall  f1-score   support

       angry       0.79      0.77      0.78       139
        calm       0.79      0.87      0.83       168
     disgust       0.75      0.70      0.73       162
     fearful       0.68      0.72      0.70       138
       happy       0.67      0.68      0.67       157
     neutral       0.73      0.65      0.69        78
         sad       0.73      0.70      0.71       163
   surprised       0.66      0.67      0.67       147

    accuracy                           0.73      1152
   macro avg       0.73      0.72      0.72      1152
weighted avg       0.73      0.73      0.73      1152
```

Fig. 18. Classification Report on the Entire Dataset

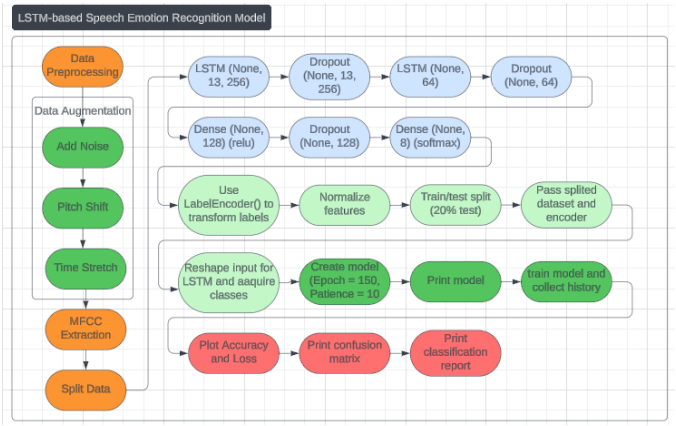

Fig. 19. The Full project Architecture

Summary Table:

	File	Actual Emotion	Predicted Emotion	Result
0	./RAVDESS/Actor_02/03-01-01-01-01-01-02.wav	neutral	happy	Incorrect
1	./RAVDESS/Actor_02/03-01-02-02-01-01-02.wav	calm	happy	Incorrect
2	./RAVDESS/Actor_02/03-01-03-01-02-01-02.wav	happy	happy	Correct
3	./RAVDESS/Actor_03/03-01-04-01-01-01-03.wav	sad	sad	Correct
4	./RAVDESS/Actor_03/03-01-05-02-01-01-03.wav	angry	angry	Correct
5	./RAVDESS/Actor_04/03-01-06-01-02-01-04.wav	fearful	happy	Incorrect
6	./RAVDESS/Actor_04/03-01-07-02-02-01-04.wav	disgust	surprised	Incorrect
7	./RAVDESS/Actor_05/03-01-08-01-01-01-05.wav	surprised	surprised	Correct
8	./RAVDESS/Actor_06/03-01-02-01-02-01-06.wav	calm	calm	Correct
9	./RAVDESS/Actor_06/03-01-03-02-01-01-06.wav	happy	happy	Correct
10	./RAVDESS/Actor_07/03-01-04-01-01-01-07.wav	sad	calm	Incorrect
11	./RAVDESS/Actor_07/03-01-05-01-02-01-07.wav	angry	angry	Correct
12	./RAVDESS/Actor_08/03-01-06-02-01-01-08.wav	fearful	happy	Incorrect
13	./RAVDESS/Actor_08/03-01-07-01-01-01-08.wav	disgust	disgust	Correct
14	./RAVDESS/Actor_09/03-01-08-02-01-01-09.wav	surprised	fearful	Incorrect
15	./RAVDESS/Actor_09/03-01-01-01-02-01-09.wav	neutral	sad	Incorrect
16	./RAVDESS/Actor_10/03-01-02-02-02-01-10.wav	calm	sad	Incorrect
17	./RAVDESS/Actor_10/03-01-03-01-01-01-10.wav	happy	disgust	Incorrect
18	./RAVDESS/Actor_11/03-01-04-02-01-01-11.wav	sad	sad	Correct

Fig. 20. Example Inference Application

IV. Conclusion

Speech emotion recognition using machine learning have various impactful applications that includes personalized human-computer interactions on the user end, help 911 dispatchers infer the emotional status of callers that may be calling under threat, and tailoring gaming experience for gamers based on their inferred emotional status. For example, a voice-based game like Valorant may implement models that immediately ban angry player for misbehaving by inferring their angry status through their voice. In this project an LSTM based model with early stopping and data augmentation was used to classify emotions into 8 different categories using the audio's extracted MFCC features that captures frequency content of audio in a similar way to human audio perception. The model was trained and iteratively refined to reach validation accuracy of 73%, exceeding the 70% accuracy goal set for this project. The final model was saved as .keras file that can be easily loaded into any script and used to infer the emotion from the MFCC features extracted from an audio file. The LSTM model was proven to be accurate enough for general inference, but a personalized model, especially for gaming or personalized human-computer interaction applications, can yield even better results by allowing the model to learn and over fit its weights for a specific user during run-time.

References

[1] "The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS)" by Livingstone & Russo is licensed under CC BY-NA-SC 4.0."The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS)" by Livingstone & Russo is licensed under CC BY-NA-SC 4.0.

[2] H. Ma, J. Wang, H. Lin, B. Zhang, Y. Zhang and B. Xu, "A Transformer-Based Model With Self-Distillation for Multimodal Emotion Recognition in Conversations," in IEEE Transactions on Multimedia, vol. 26, pp. 776-788, 2024, doi: 10.1109/TMM.2023.3271019.

[3] S. E. Eskimez, K. Imade, N. Yang, M. Sturge-Apple, Z. Duan and W. Heinzelman, "Emotion classification: How does an automated system compare to Naive human coders?," 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Shanghai, China, 2016, pp. 2274-2278, doi: 10.1109/ICASSP.2016.7472082.

[4] Tzinis, E., Paraskevopoulos, G., Baziotis, C., & Potamianos, A. (2018). Integrating recurrence dynamics for speech emotion recognition. Proceedings of Interspeech 2018, 927–931.