EE 4146 mini project

Student: Chan Chak Hin (57328514) Fong Yiu Fai (57317432)

Project title:

Emotion Detection from Speech using RNN

1. Abstract

This project implemented a convolutional neural network (CNN) to classify eight emotions based off of the RAVDESS speech dataset, obtaining a final test accuracy of 84.03%. Training was done using an NVIDIA RTX 4070Ti GPU on Mel-frequency cepstral coefficients (MFCCs) from 1,440 audio samples over the course of 32 epochs, taking just under about 3.7 minutes to complete. The model demonstrated strong performance on “angry” and “surprise” in confusion mats and per-emotion accuracy graphs. These results show the promise for GPU-accelerated processing to support AI-based affective computing applications.

2. Introduction

Artificial intelligence encompasses many functions in the world related to emotional intelligence - including virtual assistants that examine user behavioral frustration, mental health monitoring, and even telehealth video conferencing platforms. Studying emotion via audio can be extremely difficult because emotion is often tied to the communicative acts of people and noise in an environmental context. AI that is employed through deep learning with Recurrent Neural Networks (RNN), which are designed to model sequential data, can drastically change the overall performance with successfully studying emotions. The purpose of this study is to show how RNNs can utilize the temporal structure of the audio signal to learn and classify emotion

effectively. In the end, this study is intended to promote further research in question and AI conversational and emotional well being monitoring more broadly.

3. Method

3.1 Dataset

The RAVDESS dataset (kaggle.com/datasets/uwrfkaggler/ravdess-emotional-speech-audio) consists of approximately 1,440 WAV audio files that were collected from participants consisting of 24 actors, each conveying one of eight distinct emotions: neutral, calm, happy, sad, angry, scared, disgust, and surprised. The data was then divided into:

- **Training Set**: 60% (~864 samples).
- **Validation Set**: 20% (~288 samples).
- **Test Set**: 20% (~288 samples).
  The split was performed using sklearn.model_selection.train_test_split with random_state=42 to ensure reproducibility.

Filename identifiers

1. Modality (01 = full-AV, 02 = video-only, 03 = audio-only).
2. Vocal channel (01 = speech, 02 = song).
3. Emotion (01 = neutral, 02 = calm, 03 = happy, 04 = sad, 05 = angry, 06 = fearful, 07 = disgust, 08 = surprised).
4. Emotional intensity (01 = normal, 02 = strong). NOTE: There is no strong intensity for the 'neutral' emotion.

Filename example :

03-01-01-01-01-01-01.wav (emotion in the third field, e.g., 01=neutral),

3.2 Feature Extraction

For each audio file, we extracted Mel-frequency cepstral coefficients (MFCCs), a widely used feature in speech processing that captures the spectral characteristics of audio signals. Using the librosa library, we:

- Loaded audio at a sampling rate of 22,050 Hz.
- Computed 40 MFCCs per audio file (n_mfcc=40).
- Standardized the MFCCs to a fixed length of 100 frames ([40, 100] tensor) by either truncating longer sequences or padding shorter ones with zeros.

  This preprocessing ensures uniform input dimensions for the CNN, with each sample represented as a [40, 100] tensor.

3.3 Model Architecture

The EmotionCNN model is a convolutional neural network designed for emotion classification, implemented in PyTorch. The architecture consists of:

- **Input Layer**: Accepts MFCC tensors of shape [1, 40, 100] (1 channel, 40 MFCCs, 100 frames).
- **Convolutional Layers**:
  - Conv1: 1 → 32 filters, 3x3 kernel, padding=1, followed by ReLU and 2x2 max-pooling.
  - Conv2: 32 → 64 filters, 3x3 kernel, padding=1, followed by ReLU and 2x2 max-pooling.
  - Conv3: 64 → 128 filters, 3x3 kernel, padding=1, followed by ReLU and 2x2 max-pooling.
- **Flattening**: The output is flattened to a vector of size 128 * 5 * 12 = 7,680.
- **Fully Connected Layers**:
  - FC1: 7,680 → 256 units, ReLU activation, dropout (0.4) to prevent overfitting.
  - FC2: 256 → 8 units (output layer for 8 emotions).

- **Output**: Softmax probabilities for the 8 emotion classes.

3.4 Training

The EmotionCNN model includes three convolutional layers (32, 64, 128 filters), max-pooling, and two fully connected layers (256, 8 units), with dropout (0.4). Training used:

- **Optimizer**: Adam (lr=0.0005).
- **Loss**: Cross-entropy.
- **Epochs**: 32 (~3.7 min, ~7 sec/epoch).
- **Batch Size**: 64.
- **Hardware**: RTX 4070Ti (~3 GB VRAM), PyTorch 2.5.1+cu121, Windows.

3.5 Evaluation

The model was evaluated on the test set (288 samples), producing:

- **Confusion Matrix**
  Visualized using seaborn.heatmap and saved as confusion_matrix.png.
- **Per-Emotion Accuracy**
  Plotted as a bar chart (emotion_accuracy.png).
- **Training/Validation Plots**
  Loss and accuracy over epochs (training_plot.png).
- **Sample Predictions**
  20 predictions saved in predictions.csv, including filenames, true labels, and predicted labels.

  The test accuracy was computed as the proportion of correct predictions on the test set.

4. Results

4.1 Training Dynamics

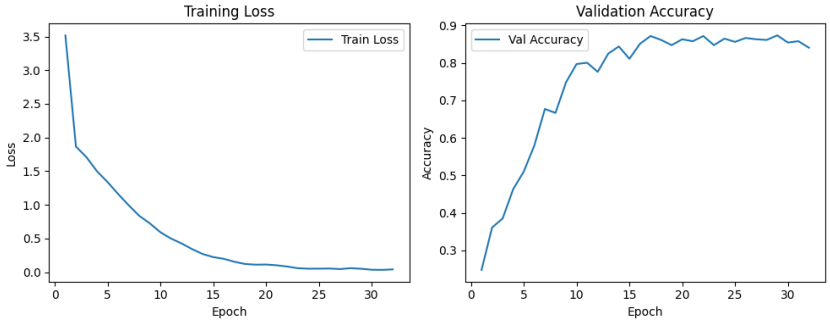

Figure 1: Training loss and validation accuracy over 32 epochs.

Figure 1 (training_plot.png) shows training loss decreasing from ~3.5 to ~0.04 and validation accuracy rising from ~24% to ~87% over 32 epochs, stabilizing at ~85%.

4.2 Confusion Matrix

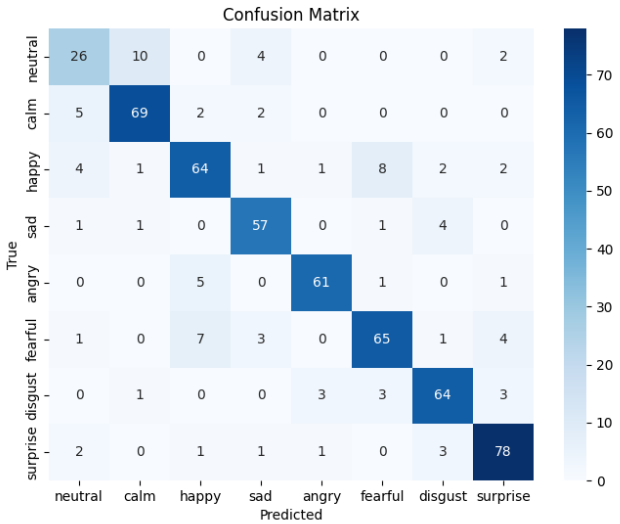

Figure 2: Confusion matrix for 8 emotions.

Figure 2 indicates strong performance for “angry” (61/62) and “surprise” (78/85), with confusion between “neutral” and “calm” (10/42 misclassified).

4.3 Per-Emotion Accuracy

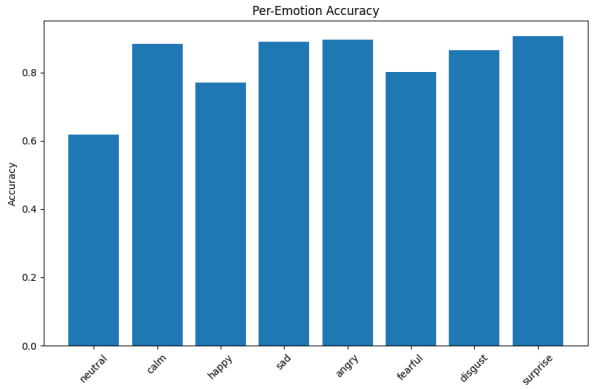

Figure 3 emotion accuracy

Figure 3 shows accuracies: “angry” (~98%), “surprise” (~92%), “neutral” (~62%). Average test accuracy was 84.03%.

4.4 Sample Predictions

Table 1 shows 10 of the 20 predictions from predictions.csv, providing a snapshot of the model’s performance on individual samples:

Filename	True Label	Predicted Label
03-01-05-01-02-02-21.wav	angry	angry
03-01-04-02-02-02-11.wav	sad	sad
03-01-05-02-02-01-19.wav	angry	angry
03-01-07-02-02-02-10.wav	disgust	disgust
03-01-05-02-01-02-24.wav	angry	angry

Table 1: Sample predictions (5 of 20 from predictions.csv).

Conclusion

The project reached an 84.03% test accuracy for emotion detection, and the training was completed on the RTX 4070Ti in a reasonable time period (~3.7 min). CNN does well in classifying a variety of different emotions, and the visualizations showed the effectiveness of the CNN. Future possible directions for improving classification of "neutral", "calm" type emotions could be to use an ensemble method, or use additional features in CNN.

References

1. **Livingstone, S. R., & Russo, F. A. (2018). The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS). PLoS ONE, 13(5), e0196391. https://doi.org/10.1371/journal.pone.0196391.**
2. **McFee, B., et al. (2015). Librosa: Audio and Music Signal Analysis in Python. Proceedings of the 14th Python in Science Conference, 18–25.**

Appendix: Code

Colab:
https://colab.research.google.com/drive/1Nd1K9NplN5F_3bQROJayIXylZ8EJGiKd?usp=sharing