Automatic Speech Emotion Recognition with Machine Learning Methods

Feiyang Yang, Zhiyu Bao

Institute of Natural Language Processing, University of Stuttgart, Germany

st196012@stud.uni-stuttgart.de, st193913@stud.uni-stuttgart.de

1. Introduction

1.1. Motivation for topic selection

Speech emotion recognition (SER) has been a concerned topic both in academia and industry. Speech, as the most natural and direct way of communication for humans, contains rich but abstract emotional information.

Moreover, both members of the group did not have practical experience in handling audio data before this lab project. Therefore, we see this project as a chance to collaboratively explore audio task and obtain hands-on experience.

To demonstrate the practical applicability of our work, we implemented a real-time web application. Nevertheless, as the primary focus of this report lies in the model design and evaluation, the web component will be discussed only briefly.

1.1.1. Repository

The complete source code and implementation details for this project are documented and available at the following GitHub repository: https://github.com/amnesiackid/automatic-speech-emotion-recognition-on-ravdess.git

2. Related Works

The task of Speech Emotion Recognition (SER) has been explored with a wide variety of methods, ranging from traditional machine learning algorithms to complex deep neural networks. This project has chosen a 1D Convolutional Neural Network (CNN) for the baseline model and pretrained Wav2Vec2 and Hubert model for fine-tuned advanced models.

2.1. Traditional classifiers

Support Vector Machines (SVMs), Gaussian Mixture Models (GMMs), and Hidden Markov Models (HMMs) are fundamentally different from our approach as their performance is highly dependent on extensive, manually engineered acoustic features.[1]

Our work intentionally diverges from this path by employing a deep learning model. The 1D CNN has the distinct advantage of automatically learning a hierarchy of discriminative features directly from a more basic feature sequence like Mel-frequency cepstral coefficients (MFCCs), thereby avoiding the complex and potentially suboptimal process of manual feature selection.

2.2. 2D or 3D CNNs on spectrograms

This technique treats the SER task as an image classification problem, feeding 2D representations of the audio signal into the network. Our work differs by processing the features as a 1D sequence rather than a 2D image. We chose this 1D route primarily for efficiency and directness.

A model of 3D CNN which operates on a three-dimensional log-Mel spectrum and introduces a ”relational perceptual self-attention mechanism”, achieving a high accuracy on emotion recognition tasks. In addition, some work has innovated in feature processing, such as using k-means clustering to determine the most discriminating ”keyframes” after extracting 88-dimensional feature vectors containing pitch, intensity, and MFCC, and then feeding these frames into the classifier.

2.3. RNN or LSTM

Recurrent neural networks (RNNs) and Long short-term memory(LSTM) are good at modeling the long-term temporal dependencies in speech. However, RNN models are computationally expensive to train, compared to CNN models. We acknowledge the potential of RNN and LSTM models in terms of performance; however, this project adopts a purely CNN-based architecture for the sake of clarity and interpretability. In a purely CNN model, we can more directly analyze the role of convolutional layers in capturing local acoustic patterns. Therefore, we first focus on optimizing a simple and efficient architecture, which lays a solid foundation for possible future research on more advanced models.
Moreover, its memory mechanism is outperformed by transformer-based models, which we will introduce below.

2.4. Transformer

The original transformer model is composed of two parts: encoder and decoder. The encoder receives an input, and builds a representation of it (its features). This part of the model is trained to acquire understanding from the input. The decoder uses the encoder’s representation (the features) along with other inputs (the previously predicted tokens) to generate an output. For SER task, transformer-based models are usually encoder-only. Examples of such models are Wav2Vec2 [2], HuBERT [3] and M-CTC-T.

Connectionist Temporal Classification (CTC) is a technique originally designed for automatic speech recognition (ASR). It utilizes the sequence of hidden states of a encoder-only model, mapping it to characters of the alphabet. In SER task, we take a CTC model and transform it into an emotion classifier by changing the labels to emotions.

3. Methods

3.1. Baseline model

3.1.1. Data preprocessing

Since models cannot directly understand wav or mp3 files, it is necessary to change the audio into a format that machine can understand. We first need to convert the audio waveform into a structured digital representation, which is also called feature extraction.

First our core task was to traverse all the audio files of the RAVDESS dataset [4] and parse out the emotional information from the file names.

- Create two empty lists 'file emotion' and 'file path', which are used to store the emotion tags and full paths corresponding to each file respectively. Through a loop, going through the folders of each actor and traverses every audio file in the current actor folder (the value of f is the file name, e.g. 03-01-06-01-02-01-12.wav).
- According to the naming rules for the RAVDESS dataset, the third element in the list (part[2]) represents emotion. Extracted here (e.g. '06'), converted into an integer 6 and added to the 'file emotion' list.

3.1.2. Data Augmentation

For an audio file, we try four different augmentation methods(noise, stretch, pitch, and shift) and choose two of them(noise and pitch). Finally we generate 4 versions of feature data(normal, pitch, noise, pitch and noise) to expand the dataset and improve the generalization ability of the model.

3.1.3. Feature Extraction

The process involves using the Librosa library to compute three key audio features for each audio file: Zero-Crossing Rate (ZCR), Root Mean Square Energy (RMSE), and Mel-Frequency Cepstral Coefficients (MFCCs). These extracted features are then organized into Pandas DataFrames for analysis in next step. Finally, the ZCR, RMSE, and MFCC features are combined into a single one-dimensional array using a horizontal stack, which serves as the input for the baseline model.

3.1.4. Model Architecture

We implemented a 1D CNN using TensorFlow/Keras. The architecture is designed to hierarchically extract features from the input sequence. After a great deal of attempt, the structure of the baseline model is followed.[5]

Convolutional Blocks: Three sequential blocks are the building block of the model. The blocks are comprised of:

1. A Conv1D layer to act as a feature detector, scanning for local patterns across the time steps. The number of filters decreases through the blocks (128 → 64 → 32), forcing the model to distill the most salient information.
2. A BatchNormalization layer in order to stabilize training and speed convergence.
3. A ReLU activation function to add non-linearity.
4. A MaxPooling1D layer to halve the sequence length, making the learned representations less sensitive to temporal shifts.
5. A Dropout layer to avoid overfitting, make the model more robust.

Aggregation Layer: A GlobalAveragePooling1D layer is applied after the final convolutional block. This layer transforms the feature sequence of shape (sequence length, 32) into a single fixed-size feature vector of shape (32,) by averaging across the time dimension. This creates a holistic summary of the features present in the entire utterance.

Classification Head: A fully-connected network with two Dense layers. The output Dense layer contains 8 units with softmax activation, making it output a probability distribution over the 8 emotion classes.

The model was trained using the Adam optimizer and the weighted categorical cross-entropy loss function.

3.1.5. Optimization

Because the performance of the baseline model was not very good, we tried different model structures, such as a two-layer CNN, a three-layer CNN, and a five-layer CNN. The results showed that the three-layer model was the most efficient. In addition, we added more datasets (CREMA-D, TESS, SAVEE), which improved the accuracy of the model from 46.2 to 56.7.

3.2. WAV2VEC2 model

For the advanced model, what we initially wanted to do is to fine-tune a pretrained model; However, we encountered unexpected challenge due to our lack of experience. In the end, we came up with an in-between idea, which is utilizing Wav2Vec2 as a feature extractor.

3.2.1. Feature Extraction

As we introduced in **2.4 Transformer** section, in the SER task, encoder-only models map the sequence of hidden states to labels. Therefore the last hidden layer of such models contains information easier to interpret for models than raw waveforms or spectrograms.

Hence, we feed waveform to pretrained Wav2Vec2 model, saving its last hidden layer as features.

3.2.2. Model Architecture

In this scenario, what the model does is akin to the last linear layer which projects the acoustic features to the labels. Nevertheless, the model is not pretrained on large data, also not compatible with Wav2Vec2 model as a CTC model is. As a result, we can not rely on a simple linear model to do the heavy lifting for us.

At this stage, we adopt a similar CNN architecture as we utilized in the baseline model. The main difference is that we significantly decrease the dropout rate. Because we do not introduce noise as we do for the baseline model.

The model receives features obtained from Wav2Vec2, and maps them to 8 emotion labels.

3.2.3. Limitations

This unusual method has obvious shortcomings. First, the model is "disabled". It relies on features extracted by a pretrained model, and this is usually not available in an audio dataset. Consequently, to use the model, we have to use Wav2Vec2 to extract features every time when we use the model. It also causes higher computational cost. In addition, while the feature extractor is pretrained, the CNN model is not. This incompatibility may affect the performance of the model.

In conclusion, this model has various shortcomings; however, this was the only option left for us after many failures we had with finetuning a model from scratch.

3.3. Finetuned Hubert model

We have been constantly improving our project. To overcome the shortcomings of the "disabled" Wav2Vec2 model, we have learned more about transformer libraries and in the end successfully finetuned a distilled Hubert model.

3.3.1. Data preprocess

First of all, we resample our audio to 16000 Hz as Hubert requires. Subsequently, we feed the waveform to pretrained Hubert feature extractor. It returns processed features and attention mask, which tells the model which part to attend to.

3.3.2. Model architecture

Encoded dataset is subsequently fed to pretrained a Hubert model. It utilizes a learning rate of *5e-5*, warmup ratio 0.1, dropout rate 0.1. The training curves demonstrate a successful training process.

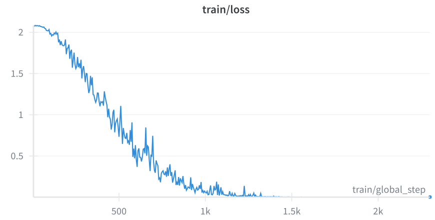

Figure 1: *Training Loss Curve*

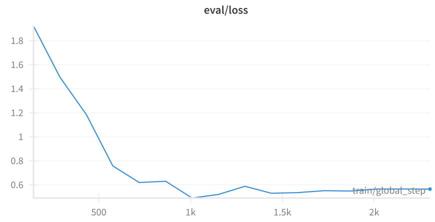

Figure 2: *Evaluation Loss Curve*

3.3.3. Limitations

Despite significantly outperforming the former models, this Hubert model still shows limitations.

This model is trained only on the Ravdess dataset, which includes exclusively audio by actors with North American accent. When we evaluate it on other audio dataset with non North American accent, such as CREMA-D [6], it shows insufficient ability to generalize. The model works satisfyingly solely on North American accent speech.

4. Findings

Our experiments yielded several key quantitative and qualitative results.

4.1. Comparison

Analysis of Results: A review of the confusion matrix revealed distinct patterns in classification performance.

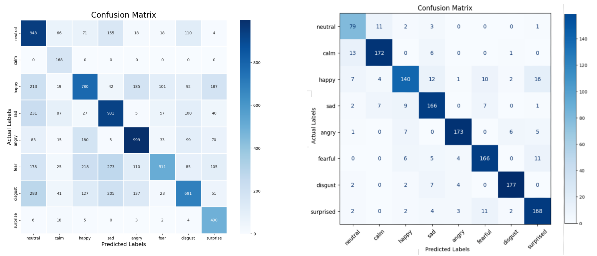

Figure 3: *Baseline model*

Figure 4: *Wav2Vec2 model*

Figure 2. Confusion matrices for two models

Model	Best accuracy
Baseline model	56.7%
Wav2Vec2 integrated model	67.2%
Finetuned Hubert model	**86.8%**

Table 1: *Models' performance on evaluation data*

4.2. Analysis of Results

The following limitations and difficulties of the SER task are commonly faced by all the models:

- Emotions with high arousal and distinct energy contours, such as angry, are classified with high accuracy.
- On the contrary, emotions with low arousal are often misclassified.
- Emotions sharing similar acoustic features that even human cannot confidently classify, such as happy and surprised, neutral and calm, are often confusing for the model.

In terms of inter-model comparison. These results demonstrate that finetuned pretrained models exceed the performance of baseline model.

An 8-emotion classification task is challenging because of the aforementioned difficulties. The emotion labels are sometimes subjective — individuals may give different labels to the same speech. Consequently, despite our effort on processing and modifying model architecture, the model performs unsatisfyingly.

The Wav2Vec2 integrated model achieves better accuracy. The pretrained model has been trained on massive audio data, therefore it has better capacity to extract audio features. However, the incompatibility between the pre-trained feature extractor and the classification model hinders the further improvement of the model. Moreover, its "disabled" nature that it relies on a separate feature extractor impedes its practical application.

By utilizing pretrained model and addressing the incompatibility issue, the finetuned Hubert model significantly outperforms the other two models. However, its performance can still be further improved by being trained on diverse datasets.

5. Achievements

It was implemented based on the plan mapped out in the preliminary proposal, with all important deadlines fulfilled.

1. Literature Review (Complete): We undertook an exhaustive search of current SER methods in order to guide our choice of architecture and chose the RAVDESS dataset as an adequate and balanced baseline against which our experiments should be conducted.
2. Data Preprocessing Pipeline (Achieved): A robust and reusable Python script was developed to automate the loading, resampling, and extraction of MFCCs and other features from the entire dataset, preparing it for model ingestion.
3. Baseline Model Implementation and Training(Reached): The 1D CNN architecture suggested, detailed in the Methods, was implemented with success utilizing the TensorFlow/Keras.
4. Advanced Model Implementation (Reached): Finetuned pretrained Wav2Vec2 and Hubert model on the Ravdess dataset, achieving better performance than the baseline model.
5. Training and Testing (Done): The neural network has been trained well with the available data. We performed hyperparameter search and conducted the final test set evaluation, obtaining the numeric results reported in the Results section.
6. Analysis and Reporting (Achieved): We performed a detailed analysis of the model's performance, including a confusion matrix review, and compiled all methods, results, and conclusions into this final report.
7. Web Application (Done): We built an interactive website to apply models to verify the practical application value of our research. After the user records audio through the front-end interface, it is sent to the server. On the server side, an advanced feature extractor converts raw audio data into valid features and hands them to the core's 1D CNN classifier for analysis. The classifier is able to accurately identify the emotions contained in speech and ultimately feed back results like ”happy” to the user, enabling smooth, real-time human-computer emotional interaction.

6. References

[1] A. S. M and A. U. M, “Enhancing speech emotion recognition through advanced feature extraction and deep learning: A study using the ravdess dataset,” in *2024 IEEE International Conference on Information Technology, Electronics and Intelligent Communication Systems (ICITEICS)*, 2024, pp. 1–7.

[2] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, “wav2vec 2.0: A framework for self-supervised learning of speech representations,” 2020. [Online]. Available: https://arxiv.org/abs/2006.11477

[3] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, “Hubert: Self-supervised speech representation learning by masked prediction of hidden units,” 2021. [Online]. Available: https://arxiv.org/abs/2106.07447

[4] S. R. Livingstone and F. A. Russo, “Ravdess emotional speech audio,” 2019. [Online]. Available: https://www.kaggle.com/dsv/256618

[5] E. Lakomkin, M. A. Zamani, C. Weber, S. Magg, and S. Wermter, “On the robustness of speech emotion recognition for human-robot interaction with deep neural networks,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 854–860.

[6] H. Cao, D. Cooper, M. Keutmann, R. Gur, A. Nenkova, and R. Verma, “Crema-d: Crowd-sourced emotional multimodal actors dataset,” *IEEE transactions on affective computing*, vol. 5, pp. 377–390, 10 2014.