DeepEmoNet: Building Machine Learning Models for Automatic Emotion Recognition in Human Speeches

Tai Vu
Department of Computer Science
Stanford University
taivu@stanford.edu

Abstract

Speech emotion recognition (SER) has been a challenging problem in spoken language processing research, because it is unclear how human emotions are connected to various components of sounds such as pitch, loudness, and energy. This project aims to tackle this problem using machine learning. Particularly, we built several machine learning models using SVMs, LTSMs, and CNNs to classify emotions in human speeches. In addition, by leveraging transfer learning and data augmentation, we efficiently trained our models to attain decent performances on a relatively small dataset. Our best model was a ResNet34 network, which achieved an accuracy of 66.7% and an F1 score of 0.631.

1 Introduction

In recent decades, the advent of machine learning technologies has accelerated research in spoken language processing. In particular, the applications of neural network architectures like CNNs (LeCun et al., 1995), LSTMs (Hochreiter and Schmidhuber, 1997), and Transformers (Vaswani et al., 2017) have led to major advancements and desirable outcomes in automatic speech recognition and speech synthesis programs.

One interesting area of spoken language research is speech emotion recognition (SER). This problem involves classifying emotions like "happiness" or "anger" based on audio clips of human speeches. This is a highly important task, because enabling computers to understand human emotions can help facilitate communication between humans and machines. However, while there has been significant research in building AI-powered emotion detection systems, closing the gap between AI performance and human performance still proves to be challenging, due to the ambiguity and complexity of human emotions.

Therefore, in this project, we developed several machine learning models that utilized SVMs, CNNs, and LTSMs for automated emotion classification in human speeches. We also implemented transfer learning and data augmentation techniques during the training process, which allowed our models to achieve good performances with little training data.

2 Related Works

Over the last few years, there have been a increasing number of studies on speech emotion recognition (SER). For instance, Schuller et al. (2003) leveraged a Hidden Markov Model (HMM) to extract features from speech signals and used them to detect emotions. More recent research has utilized Mel Frequency Cepstral coefficients (MFCCs), which has proven very useful in automatic speech recognition. Specifically, Demircan and Kahramanli (2018) extracted MFCCs from the EMO-DB dataset, and then combined them with fuzzy C-means clustering and k-nearest neighbors (kNN) for emotion prediction.

Meanwhile, together with recent breakthroughs in deep learning, many studies have focused on leveraging the power of neural networks for emotion classification. Particularly, Lim et al. (2016) applied CNN and LSTM network layers on top of short-time Fourier transform representations of the EMO-DB raw audio data. This approach demonstrated great improvements in predictive accuracy over traditional classification methods. However, most of those deep learning based systems required a large amount of training data in order to achieve high performances. Our project was different, because we trained our machine learning models on a relatively small database. We will demonstrate that incorporating data augmentation and transfer learning can effectively enable our systems to over-

come the lack of data, address overfitting issues, and attain decent performances.

In addition, a number of studies have focused on building multimodal systems that harness additional information from videos or texts to improve speech emotion classification. For example, Kim et al. (2013) combined hand-crafted speech features such as pitch, energy, and mel-frequency filter banks (MFBs) with facial landmark features from videos. On the other hand, Tzirakis et al. (2017) leveraged 1D convolutional layers to encode features from speeches, while using ResNet50 to extract visual information from video frames. The combined features were passed through an LSTM module to perform final prediction. While this multimodal approach led to some improvements in accuracy levels, it is crucial to note that visual and textual information is not always available. Therefore, building audio-only emotion detection systems is highly important for use cases where we only have audio data. This insight motivated us to develop and train our machine learning models to output correct emotion labels solely based on input audio clips without using any visual or textual data.

3 Approach

3.1 Models

Our machine learning system included an encoder, which was followed by a classifier. The encoder received an audio clip and then produced a vector representation of the input data. Subsequently, this encoding was fed into the classifier, which outputted an emotion label.

Model 1: MFCC and SVM

As a starting point, we implemented the feature extractor using the Mel Frequency Cepstral Coefficients (MFCC). Afterwards, we took the averages of these MFCC input features across the time dimension and then used them to train a Support Vector Machines (SVM) model (Boser et al., 1992) to classify different emotions.

Model 2: Log mel spectrograms and LSTM

Our second model encoded each data point by computing a mel-scaled spectrogram and then converting it to log space. We built an LSTM neural network as our classifier. This network contained 2 bidirectional LSTM layers, followed by a dropout layer, a linear layer, and a softmax layer.

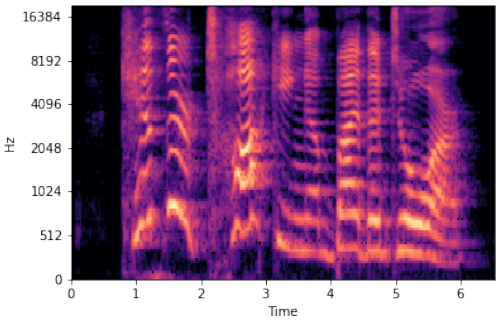

Figure 1: Log mel spectrogram features of an example.

Model 3: Log mel spectrograms and CNN

In this model, we also extracted log-scaled mel spectrograms for the input speech data. Since these features were similar to 2D image arrays (shown in Figure 1), we then fed them into a CNN classifier in order to obtain emotion labels. Previously, we intended to put raw waveforms directly through the CNN model. However, during our experiments, we found out that training the CNN on log-scaled mel spectrograms was easier and more stable.

We chose ResNet34 (He et al., 2016) as our CNN architecture. Additionally, we experimented with two different approaches: training a ResNet34 network from scratch and using transfer learning to finetune a ResNet34 model that was pretrained on the ImageNet database (Russakovsky et al., 2015).

3.2 Data Augmentation

As we developed and trained our models on a small speech dataset, data augmentation would be helpful in generating more training data and dealing with overfitting problems.

Image-based Data Augmentation

In particular, since our CNN models were trained on image-like 2D arrays of log-scaled mel spectrograms, we applied several data augmentation methods on these input data, which include rotating by a small degree, zooming in, and changing brightness. Although such image-based augmentation techniques were more common in computer vision tasks and were not directly applied to audio data, we will demonstrate in Section 5.4 that these techniques indeed helped prevent overfitting and improve model performance.

Progressive Resizing

Another augmentation method that we used was progressive resizing (Colangelo et al., 2021).

Specifically, we first trained the CNN models on smaller versions of the log-scaled mel spectrogram arrays (128×128), and then finetuned the networks on arrays of larger sizes (256×256). This approach not only augmented the training data, but also allowed the models to train much faster.

Mixup

In addition, we harnessed Mixup, a data augmentation technique that generated convex combinations of pairs of training examples and their labels (Zhang et al., 2018). Particularly, for two randomly sampled data points (x_i, y_i) and (x_j, y_j), this method constructed a new example of the form:

$$\begin{aligned} \tilde{x} &= \lambda x_i + (1-\lambda)x_j \\ \tilde{y} &= \lambda y_i + (1-\lambda)y_j \end{aligned}$$

Here, x_i, x_j are input vectors, y_i, y_j are one-hot label encodings, and $\lambda \in [0, 1]$. In this way, Mixup acted as a regularizer that encouraged the linear behaviors of the models, reduced their variance, and enhanced their generalization powers.

4 Implementation

I implemented the code for this project in Python using PyTorch (Paszke et al., 2019), FastAI (Howard and Gugger, 2020), Scikit-learn (Pedregosa et al., 2011), Librosa (McFee et al., 2015). All the code can be found here.

5 Experiments

5.1 Data

In this project, we used the Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) database (Livingstone and Russo, 2018) and the Surrey Audio-Visual Expressed Emotion (SAVEE) database (Jackson and Haq, 2014). We combined them into a single dataset for training and testing our models.

RAVDESS is an English language database that contains 1440 utterances. This dataset was made by 24 actors (12 female and 12 male), who said two sentences "Kids are talking by the door" and "Dogs are sitting by the door" with various emotions. Meanwhile, the SAVEE database consists of 480 audio clips created by 4 male actors, and each of them recorded 15 sentences. There are 8 different emotion classes, including `neutral`, `calm`, `happy`, `sad`, `angry`, `fearful`, `disgust`, and `surprised`.

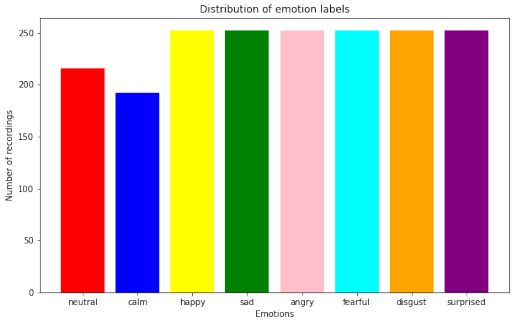

Figure 2: Distribution of emotion labels in the dataset.

The duration of each utterance ranges from 3 to 5 seconds. The total duration of audio recordings is roughly 2 hours. In addition, we can see in Figure 2 that most of the emotional classes are relatively well balanced. The `neutral` and `calm` labels contain slightly fewer audio clips than the other 6 classes.

We split the dataset into 90% for training, 5% for validation, and 5% for testing.

5.2 Experiment Details

For the SVM model, we produced 20 MFCCs for each input audio clip. We chose an RBF kernel for the SVM algorithm.

For the LSTM and CNN models, we generated 128 mel bands when converting input speeches to mel spectrograms. We trained the LSTM model and the vanilla CNN model (with no pretraining) for 200 epochs. Meanwhile, for the ResNet34 model that was pretrained on ImageNet, we finetuned its weights for 30 epochs. We used a batch size of 64 and a learning rate of 0.001, with a decay rate of 0.9. We trained the above neural networks using the Cross Entropy loss and the Adam optimization algorithm (Kingma and Ba, 2014).

5.3 Evaluation Methods

Since this project tackled a classification problem, we used classification accuracy scores and F1 scores for evaluating model performance.

5.4 Results

As shown in Table 1, the SVM algorithm produced an accuracy of 51.7% and an F1 score of 0.509. This result was better than we expected, because the model only took into account the mean values of the MFCC features across the time dimension. In other words, the SVM algorithm did not get access to useful temporal dependencies amongst

Models	Accuracy	F1 Scores
SVM	51.7%	0.509
LSTM	52.8%	0.497
CNN (trained from scratch)	45.8%	0.426
CNN (transfer learning)	57.3%	0.528
CNN (transfer learning, data augmentation)	**66.7%**	**0.631**

Table 1: Performance of different models on the validation set.

the input MFCC features, but still learned to predict emotions with more than 50% accuracy.

After that, the LSTM model performed slightly better than the SVM algorithm, with a higher accuracy of 52.8% and a comparable F1 score of 0.497. When investigating its training process, we can see that the performance was still quite low because the LSTM network was overfitting to the training data. In particular, the model learned to decrease training losses to a small value (around 0.5), but the validation losses were still high (around 2.9).

A similar pattern occurred for the vanilla CNN model (with no pretraining), as it only produced 45.8% accuracy. In this case, another issue is that because the training set was too small, the Resnet34 network was not able to learn good representations of the speech contents, so it could not generalize well to unseen data.

In fact, when we finetuned the ResNet34 model with pretrained weights from ImageNet, the performance went up significantly (57.3% in accuracy and 0.528 in F1 score). Therefore, we can see that the neural network learned useful feature representations of the speech data after being pretrained on a large database like ImageNet. When it was finetuned on our small dataset, the model was able to transfer its prior knowledge about images to reading and extracting information from image-like log-scaled mel spectrogram arrays. The finetuning process then helped the model to adapt to the domain of our dataset even better, which enhanced its performance.

Finally, the ResNet34 model with both transfer learning and data augmentation achieved the best performance, with an accuracy of 66.7% and an F1 score of 0.631. This illustrates the effectiveness of data augmentation techniques in boosting our model performance. Indeed, as we can see in the upper plot of Figure 3, the ResNet34 network without data augmentation was still overfitting, with low training loss values and high validation loss values. This means that the gap between the training losses and the validation losses was still very large. However, this problem was alleviated with the support of data augmentation, as shown in the lower plot of Figure 3. Both the training losses and the validation losses decreased gradually, and the gap between them was significantly narrowed.

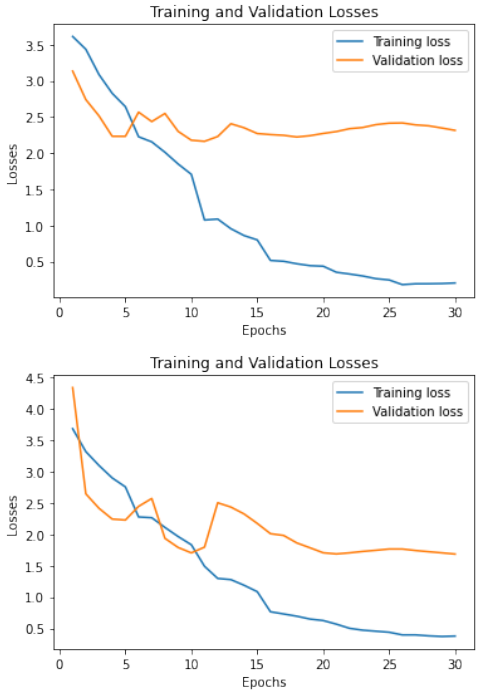

Figure 3: Training and validation losses across 30 epochs for the ResNet34 model without data augmentation (upper) and with data augmentation (lower).

Meanwhile, because the accuracy of our final model was less than 70%, there is still a lot of room for improvement. One of the main challenges faced by our models was that RAVDESS and SAVEE were two simulated datasets, which consisted of several actors repeating the same sentences with various emotions. Hence, the speech contents in these datasets were not diverse enough for our machine learning programs to learn proper representations of input audio data and detect correlations between human speeches and emotions. In addition, we can observe in Figure 4 that the

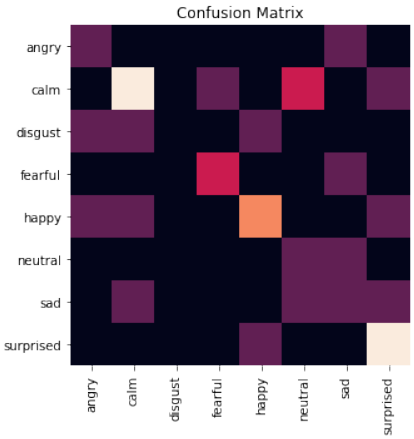

Figure 4: Confusion matrix for the best CNN model.

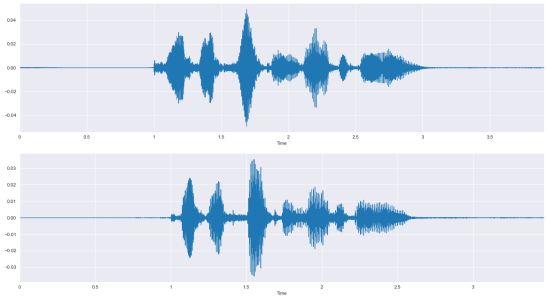

Figure 5: The waveforms of a neutral utterance (upper) and a calm utterance (lower) from the same actor.

ResNet34 model performed well on certain positive classes like `surprised`, `happy`, and `calm`, while produced lower accuracy on some other negative classes like `disgust` and `angry`. Furthermore, there was some confusion between certain pairs of emotion labels, such as `neutral` and `calm`. This issue is understandable, because the audio clips from these two classes in our dataset often sound similar. Two examples from those two classes are shown in Figure 5.

6 Conclusion

Overall, in this study, we developed a number of machine learning models, including SVMs, LSTMs, and CNNs, for inferring emotions from human speeches. Our models were trained and evaluated on small dataset created from the RAVDESS and SAVEE databases. Our best model was a ResNet34 neural network, which achieved an accuracy of 66.7% and an F1 score of 0.631. This is a promising result, given the small size of our training set. With more training data, the model will definitely be able to learn better and recognize emotion classes with higher accuracy levels. In addition, we demonstrated the benefits of transfer learning and data augmentation in boosting model performance. Particularly, transfer learning allowed the model to overcome the lack of audio data and learn good feature representations of speech contents, while data augmentation helped create more training examples, prevent overfitting issues, and enhance the robustness and generalization of the model.

The next step would be performing more hyperparameter tuning in order to improve our current models. Additionally, we are interested in experimenting with a combination of CNN or LSTM layers for better performances. Furthermore, given the great advantages of data augmentation, we want to implement several audio-based data augmentation techniques such as pitch shift, change in loudness, change in speed, and SpecAugment (Park et al., 2019), as they might be able to further reduce overfitting and generalization errors in our training pipeline. Finally, because transfer learning is also beneficial, we would like to finetune some pretrained speech models such as wav2vec (Schneider et al., 2019) and SpeechBERT (Chuang et al., 2019), and see how they perform in the speech emotion recognition task.

7 Contribution

Because this is a solo project, I (Tai Vu) implemented the entire code for the project, including data preprocessing, data pipeline, training pipeline, machine learning models, and evaluation.

References

Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. 1992. A training algorithm for optimal margin classifiers. In *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, pages 144–152.

Yung-Sung Chuang, Chi-Liang Liu, Hung-yi Lee, and Lin-shan Lee. 2019. Speechbert: An audio-and-text jointly learned language model for end-to-end spoken question answering. *arXiv preprint arXiv:1910.11559*.

Federico Colangelo, Federica Battisti, and Alessandro Neri. 2021. Progressive training of convolutional neural networks for acoustic events classification. In *2020 28th European Signal Processing Conference (EUSIPCO)*, pages 26–30. IEEE.

Semiye Demircan and Humar Kahramanli. 2018. Application of fuzzy c-means clustering algorithm to spectral features for emotion classification from speech. *Neural Computing and Applications*, 29(8):59–66.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Jeremy Howard and Sylvain Gugger. 2020. Fastai: A layered api for deep learning. *Information*, 11(2):108.

Philip Jackson and S Haq. 2014. Surrey audio-visual expressed emotion (savee) database. *University of Surrey: Guildford, UK*.

Yelin Kim, Honglak Lee, and Emily Mower Provost. 2013. Deep learning for robust feature generation in audiovisual emotion recognition. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 3687–3691. IEEE.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Yann LeCun, Yoshua Bengio, et al. 1995. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995.

Wootaek Lim, Daeyoung Jang, and Taejin Lee. 2016. Speech emotion recognition using convolutional and recurrent neural networks. In *2016 Asia-Pacific signal and information processing association annual summit and conference (APSIPA)*, pages 1–4. IEEE.

Steven R. Livingstone and Frank A. Russo. 2018. The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS). Funding Information Natural Sciences and Engineering Research Council of Canada: 2012-341583 Hear the world research chair in music and emotional speech from Phonak.

Brian McFee, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. 2015. librosa: Audio and music signal analysis in python. In *Proceedings of the 14th python in science conference*, volume 8, pages 18–25. Citeseer.

Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le. 2019. Specaugment: A simple data augmentation method for automatic speech recognition. *Interspeech 2019*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. 2015. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252.

Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. 2019. wav2vec: Unsupervised pre-training for speech recognition. *arXiv preprint arXiv:1904.05862*.

Björn Schuller, Gerhard Rigoll, and Manfred Lang. 2003. Hidden markov model-based speech emotion recognition. In *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03).*, volume 2, pages II–1. IEEE.

Panagiotis Tzirakis, George Trigeorgis, Mihalis A Nicolaou, Björn W Schuller, and Stefanos Zafeiriou. 2017. End-to-end multimodal emotion recognition using deep neural networks. *IEEE Journal of Selected Topics in Signal Processing*, 11(8):1301–1309.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762*.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. 2018. mixup: Beyond empirical risk minimization.