**CZ4042 Group Project Report**

Audio Emotion Recognition

Li Bingzi, Zeng Yanxi, Zhang Yuehan

# *Abstract-*The aim of this project is to build a deep learning model that can classify the audios into several emotions.

**Keywords:** CNN, Classification, Audio Emotion Recognition

# Introduction

The idea of this project…

# Literature view

Start…

# Methodology

## Neural Network and Keras

Keras is a well-known deep learning API in Python. It wraps frequently used methods and algorithms in deep learning to enable easy model building and access to the history of model training. It also contains some data pre-processing API, but not as many as sklearn. Therefore, sklearn will be used as well in this project alongside Keras.

## Data Pre-processing

In this project, the data is audio files in .wav format. The input of the CNN network is images. Thus a pre-processing method is needed.

**MFCC**

In this project, two audio pre-processing techniques are applied and compared. One is MFCC, which is derived from a type of cepstral representation of the audio clip. It is a time sequence with MFCC bands on the y-axis. For this dataset the bands are defined to 30.

Chart

Description automatically generated

Chart

Description automatically generated

1. Audio wave transformed into MFCC ( visualized by color bar)

**Melspectrogram**

Another is log-melspectrogram. Mel-spectrogram is less compressed than MFCC. MFCC representation is from taking the logs and compute DCT from mel-spectrogram. In this project we’ll take both method into experiment and compare the accuracies.

A picture containing monitor, computer, clock

Description automatically generated

1. The same audio wave transformed into melspec (without log transform)

## 2D CNN

When processing dataset like images, ConvNet is able to capture the Spatial and Temporal dependencies in an image through relevant filters.

In this experiment, a simple and commonly used architecture of ConvNet is designed: Conv blocks followed by fully connected layers.

The standard Conv block includes: 2DConv layer, batch normalization, MaxPooling layer and dropout.

In this experiment, there are 32 channels for each layer. Dropout is applied to the layers.

A picture containing diagram

Description automatically generated

1. Convolutional Network [from lecture slides]

## CNN + LSTM

Blahblahblah….

## CNN + Attention based LSTM

To further enhance the learning in the LSTM layer, we experimented with the attention mechanism. The attention mechanism is able to concentrate on different parts of the extracted features, which is important as the emotional saturation and contribution to the prediction outcome could vary across the audio clip. For example, the silent segments and less emotional segments of speech contain less emotional information. Hence, attention mechanism could be applied in the output of the LSTM layer to produce the weighted context vector that could pay particular attention to the more important parts of the inputs. [1]

In our work, the LSTM layer would be followed by an attention layer that produces the weighted outputs as the inputs for a dense layer before the output layer in order to extract the most useful information from the LSTM outputs for the emotion recognition.

## CNN + Unsupervised Representation Learning with Autoencoder

Other than representation of the features from supervised learning, unsupervised representation learnt from unlabeled could also improve the recognition accuracy. A similar approach has been adopted in [2] to improve a CNN-based Speech Emotion Recognition task, where an autoencoder is built on unlabeled audio clips and the learnt representation from the encoder is fed into the original CNN-based model to enhance recognition. In [3], similarly, different autoencoders were trained on different features to provide input features for emotion recognition.

In our work, an autoencoder would be trained form both the RAVDESS dataset and the CREMA-D dataset. Different model architectures and hyperparameters for the autoencoders would be experimented to find the optimal autoencoder architecture.

The learnt representation from the encoder of the autoencoder would then be concatenated to the intermediate outputs from the original recognition model. The combined representation would then be passed into a dense layer, and finally, the output layer.

# experiments and Results

## Data Exploration

**RAVDESS**

The recording of RAVDESS dataset is very clear. The upside of this dataset is that it includes both genders.

The downside is that all of the speakers vocalize the content in North American accent. And there’re only two sentences uttered. These might cause the model to fit in well in this dataset but performs worse on another dataset.

Note that surprise is discarded from the dataset. There are two reasons:

1. CREMA-D dataset does not have surprise label. If the model is built solely on “surprise” audios from RADVESS, it will bias towards this dataset.

2. “Surprise” is not a clear emotion. There could be pleasant surprise, more like “happy”. There could be unpleasant surprise, more like “disgust” or “fear”.

The dataset includes:

|  |  |
| --- | --- |
| female\_neutral | 144 |
| male\_neutral | 144 |
| female\_sad | 96 |
| male\_sad | 96 |
| female\_happy | 96 |
| male\_happy | 96 |
| female\_disgust | 96 |
| male\_disgust | 96 |
| female\_angry | 96 |
| male\_angry | 96 |
| female\_fear | 96 |
| male\_fear | 96 |

1. emotions in the RAVDESS dataset

Comparing sentences:

As shown from the figure below, different sentences uttered have different waveplots. However, the two saveplots have similar feature, which is huge spike in the sound waves, which might be a prominent feature for angry emotion. Different contents uttered does not affect the feature.

Chart

Description automatically generated

1. female, intense angry, speaking "Kids"

Chart

Description automatically generated

1. female, intense angry, speaking "Dogs"

Comparing emotions:

Comparing figure 4 and figure 5, same gender speaking same content in angry and neutral emotion. Clear feature can be distinguished. For angry emotion, there are spikes and the volume is high. Neutral voice has low volume and consistent pace.

Chart

Description automatically generated

1. female, neutral, speaking "Dogs"

Comparing genders:

Comparing figure 4 and figure 6, female and male speaking the same sentence with same emotion, similar patterns can be seen in the sharp spikes.

The feature which needs attention is that male usually has much lower pitch than female. And more interesting that female tends to express the emotion more strongly.

To prevent genders from affecting the result of emotions, emotions and genders are both in the output label in this project. Therefore, there are 14 class labels, as shown in table 1.

Chart

Description automatically generated

1. male, intense angry, speaking "Dogs"

Text, whiteboard

Description automatically generated

1. male vs female, intense angry, speaking "Dogs" (taking average across 30 mfcc bands)

**CREMA-D**

CREMA-D dataset is much larger than RAVDESS.

The downside of this dataset is that the audios are taken from movie. They are not the best of quality.

The upside is that this dataset covers various accents, genders, age. In this project, the two datasets are combined to provide a gender age and accent invariant dataset.

In the baseline 2D CNN build in the next part, it will be shown that adding CREMA-D dataset greatly reduce the model testing accuracy. However, this lower-quality data with high variety shows how well the model will perform in real life.

Comparing with RAVDESS dataset:

Comparing figure 3 and figure 8, from different sentences uttered in these two dataset, the sharp spikes in the angry emotion are still present. This is the base for the classification model to be possible.

Chart, histogram

Description automatically generated

1. female, intense angry

|  |  |
| --- | --- |
| female\_neutral | 656 |
| male\_neutral | 719 |
| female\_sad | 696 |
| male\_sad | 767 |
| female\_happy | 696 |
| male\_happy | 767 |
| female\_disgust | 696 |
| male\_disgust | 767 |
| female\_angry | 696 |
| male\_angry | 767 |
| female\_fear | 696 |
| male\_fear | 767 |

1. emotions in the combined dataset

## Baseline 2D CNN

**RAVDESS**

**Chart, line chart

Description automatically generated**

1. RAVDESS dataset only, MFCC, train & test acc plot

Model accuracy: 76.92%

**Chart, histogram

Description automatically generated**

1. RAVDESS dataset only, MFCC, train & test loss plot

**A picture containing background pattern

Description automatically generated**

1. RAVDESS dataset only, MFCC, testing result confusion matrix

**RAVDESS + CREMA**

**Chart, line chart

Description automatically generated**

1. RAVDESS + CREMA, MFCC, train & test acc plot

model accuracy: 57.71% (highest converging acc: 59%)

**Chart, histogram

Description automatically generated**

1. RAVDESS + CREMA, MFCC, train & test loss plot

**A picture containing treemap chart

Description automatically generated**

1. RAVDESS + CREMA, MFCC, confusion matrix

**MFCC vs Melspec**

Chart, line chart

Description automatically generated

1. RAVDESS + CREMA, melspectrogram, train & test acc

Model testing accuracy: 58.81%

Chart, line chart, histogram

Description automatically generated

1. RAVDESS + CREMA, melspectrogram, train & test loss

A picture containing treemap chart

Description automatically generated

1. RAVDESS + CREMA, melspectrogram, confusion matrix

## CNN + LSTM

Start

## CNN + Attention based LSTM

## Autoencoder

To select the optimal set of hyperparameters for the autoencoder model, we consider the following factors:

|  |  |
| --- | --- |
| **Modification** | **Options** |
| Model architecture | 1. DNN with Dense layer 2. RNN with LSTM layer 3. RNN with GRU layer |
| Model complexity | 1. 3-layer encoder for DNN 2. 4-layer encoder for DNN 3. 1-layer encoder for RNN 4. 2-layer encoder for RNN |
| Regularization | 1. None 2. Dropout = 0.2 |
| Encoding dimension | 1. 100 2. 300 |
| Feature extracted | 1. MFCC 2. Melspectrogram |

A batch size of 64, the Adam optimizer and the Mean Square Error loss function are used for all candidate autoencoder models.

**DNN Autoencoder with Dense Layers**

A picture containing text

Description automatically generated

1. DNN Autoencoders Test Loss

From the above plotting of DNN autoencoders’ test losses, we can conclude the following observations:

1. Adding dropouts relieves overfitting but also hurt the model performance.
2. The optimal model for DNN autoencoders is 3-layer with 300 encoding dimension without dropout.

# Discussion

## Baseline 2D CNN

**RAVDESS + CREMA vs RAVDESS only**

As shown in the experimental results, the model trained and tested using only RAVDESS dataset has a much higher accuracy (76.92%) than model trained and tested using the combined dataset (57.71%).

The reason is that RAVDESS is a relatively small dataset with only 24 actors and 2 sentences repeated several times for each emotion. The 24 actors are professional and all have North American accent. This will cause the model to remember some patterns well which do not apply to audios from different age groups and accents.

The combined dataset yield in lower accuracy but showed how the model will behave in the real world. The combined dataset has audios (1) from both genders (2) with different accents (3) from different age groups (4) recorded with background noises.

**MFCC vs Melspectrogram**

Comparing the accuracies, they are very close. Therefore, MFCC is used for the rest of the experiment.

**Analysis on the Confusion Matrix**

As shown in the confusion matrix, the most well classified categories are strong emotions, like angry, fear, sad and happy. Neutral are well classified. Others like disgust are not well classified, easy to mix with fear, happy neutral and sad.

There are some other points which deserves notice: neutral is often misclassified as sad, fear is often misclassified as sad.

Comparing male and female, the emotions from male speaker do not classify as well as female speaker.

# References

[1] Cs.toronto.edu. 2020. CIFAR-10 And CIFAR-100 Datasets. [online] Available at: <https://www.cs.toronto.edu/~kriz/cifar.html> [Accessed 1 November 2020].- sample

[2] sample