ID2223
Scalable Machine Learning and Deep Learning

Project Report

FALL 2021

Yuchen Gao
Weikai Zhou

January 8, 2022

KTH EECS Embedded Systems

1 Problem Description

We would like to predict the emotions from the voice and output the corresponding generalized emotion pictures from the facial expression dataset. For example, if you say something angrily, the program will output some angry facial expressions on the screen.

2 Dataset

We use two datasets:

- Toronto emotional speech set (TESS)[1]: A set of 200 target words were spoken in the carrier phrase "Say the word ..." by two actresses (aged 26 and 64 years) and recordings were made of the set portraying each of seven emotions (anger, disgust, fear, happiness, surprise, sadness, and neutral). There are 2800 stimuli in total. 60% of the dataset is used as training set, 20% as the validation set and 20% as the test set.
- Facial Expression Detection[2]: The data consists of 48×48 pixel grayscale images of faces. The faces have been automatically registered so that the face is more or less centered and occupies about the same amount of space in each image. The training set consists of 28,709 examples. Validation set and test set both have 3,589 examples.

3 Implementation

3.1 Audio Part

In this part, we load the Tess data set and check the number of files, then plotting the number of different categories of emotions. We load an example and play it. The spectrum of the sound file is plotted. The next step is to augment the data so that the normalization and regularization are realized. Then we load the data set and process all of the data, then storing the features as csv file. We have the one-hot encoder to process labels and scaler to process features, then the data is prepared and can be feed into the model. Up to now, the function is trained. The CNN-LSTM model is composed of 4 layers of CNN and one LSTM layer, then passing the output to Dense layer, and sending it to a 7 output units Dense layer with softmax function. The output is decoded by the same one-hot encoder. We use the ReduceLROnPlateau to reduce overfitting.

3.2 Facial Expression Part

In this part, we first split the dataset into training, validation and test according to the labels given by the dataset. We also need to covert the pixels into floating numbers ranging from 0 to 1, which will easier for use to train the model and plot the facial expression. Importantly, we need to convert emotion labels into one-hot encoding. We build the model to fit the training set, with batch_size = 64, epochs = 30. We also use EarlyStopping to prevent overfitting.

[1]Source: `https://tspace.library.utoronto.ca/handle/1807/24487`

[2]Source: `https://www.kaggle.com/shawon10/facial-expression-detection-cnn/data`

3.3 Combination Part

In this part, we first get the predicted emotions for the test set of facial expressions. Then we randomly choose an audio to predict its emotion. With this emotion, we can go through the test set of facial expression and output 5 figures that the predicted labels coincide.

4 Results

4.1 Results for Audio Part

The figure (Figure 1) below shows the accuracy and loss during the training process of the audio emotion corresponding. We use ReduceLROnPlateau to reduce overfitting and we finally get the accuracy of our model on test data around 80.1.

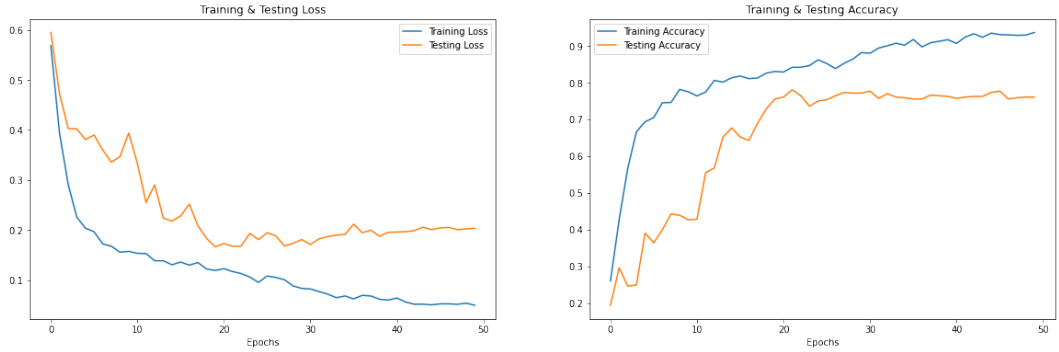

Figure 1: Loss and accuracy of training for audio

4.2 Results for Facial Expression Part

The figure (Figure 2) below shows the accuracy and loss during the training process of facial expression figures.

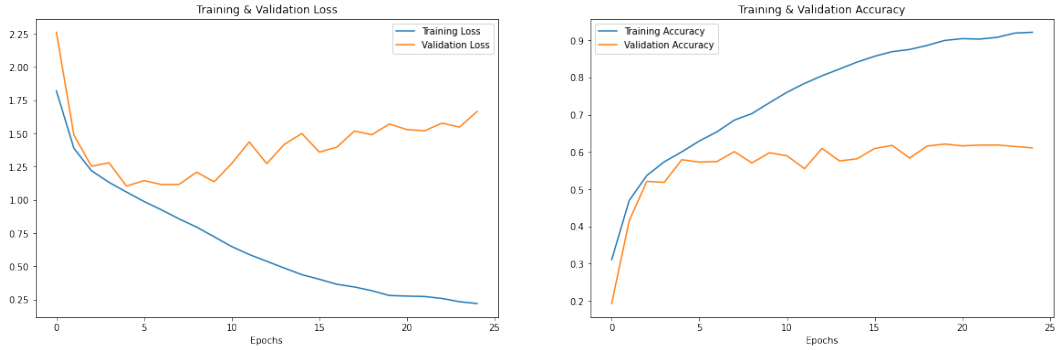

Figure 2: Loss and accuracy of training for facial expression

Although we used EarlyStopping trying to prevent overfitting, it still occurs. We may change the patience into some smaller number. And the accuracy on the test set is generally between 0.61 and 0.64, which is not good enough. Actually, the dataset we use is not good enough. The dataset is collected using crawler program and some figures are wrongly labeled or have watermarking.

Other high-quality datasets like Japanese Female Facial Expressions (JAFFE) and Extended Cohn-Kanade (CK+) are either not available or not allowed to be used in a course project.

4.3 Results for Combination Part

In order to overcome the side-effect of the low-quality datasets, we decide to output 5 facial expressions at the same time. For example, if we use "YAF_rain_happy.wav" as the input voice, we can get the following result (Figure 3).

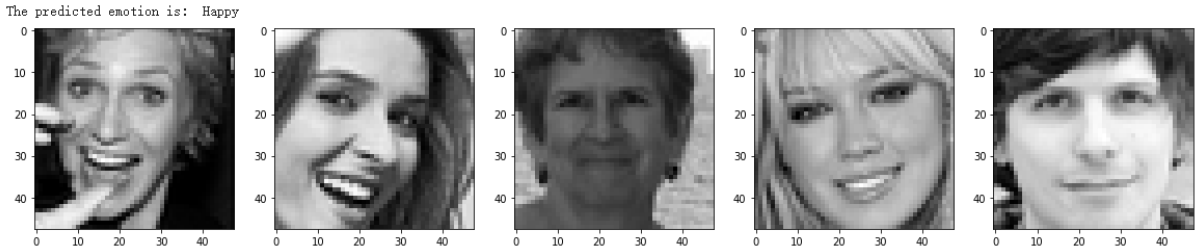

Figure 3: Final output

From the figure, we know that the prediction of the emotion of the voice is correct, which is "Happy". Besides, the facial expressions we output are generally correct.

5 Run the Code

The code is written and run on the Google Colaboratory. As long as you have the datasets in the corresponding paths in your Google Drive, it should be easy to run the code.