Audio Emotion Recognition System

Ece Omurtay [1] **Nur Altıparmak** [1]

Abstract

This paper will introduce you to the audio emotion recognition system. The aim of this project is to classify audios labeled with 8 different emotions (neutral, calm, happy, sad, angry, fearful, disgust and surprised). We work with the RAVDESS dataset where 1440 audio files in total. Odd-numbered actors are male, even-numbered actors are female. 24 actors and each has 60 different labeled audios. We use Convolutional Neural Network, LSTM, Random Forest, Decision Tree, Gradient Boost, Catboost and XGBoost to train our model.

1. Introduction

Speech recognition has become one of the most commonly used areas in technology nowadays. There are many instances of it like Apple's Siri, Microsoft's Cortana and Amazon's Alexa that were developed by well-known brands. Emotion recognition is part of this speech recognition topic.

Human beings represent their emotions via voices not only gestures and facial expressions. Emotions can be defined as people's mental state or feeling that is generated due to some interactions like communications or situations that occur in daily life. Emotions can be quite changeable from person to person and they play an important role in our daily lives. They help to understand people and give us information about people's mental health. If we can recognize and label emotions from people's voices in a fast and effective way, we can use this labeled data in many fields that people are the main subject. It can be used detecting mental state and mental health especially in psychological researches or medical experiments that people are included in. Moreover, detecting emotional states can be an indicator of whether people in safe or not. Fearful labeled emotions can be used in understanding the behaviors of a person whether he is in danger. In addition to this, it can be used to build a kind of lie detector. Sad labeled emotions can be used in suggestion systems for increasing a person's mental state and preventing depressive feelings. So, we made a system that recognizes emotions from voices of people.

In section 2, other works is referred related to our subject.

In section 3, methods that we use in this project is told.

In section 4, results of methods are told.

In section 5, conclusion and future work is referred.

2. Related Work

There are many works done by using Gaussian Mixture Model (GMM) [4], Hidden Markov Model (HMM) [1], Support Vector Machine (SVM) [3, 6], and Neural Networks (NN). CNN [5] based algorithms work very well in speech recognition systems. In project (1), HMM model was used. Overall accuracy is higher than %70. In project (2), random forest classifier was used. The best average accuracy is %86,25. In project (3), the best average accuracy is %95,1 with features such as Linear Prediction Coefficients (LPC), Mel-Frequency Cepstrum Coefficients (MFCC) and Linear Prediction Cepstrum Coefficients (LPCC). In project (4), GMM was used. The best accuracy for angry class is %100, happy is %67, sad is %89, neutral is %73 and fear is %50. In project (5), the best accuracy with using MFCC is %71,6 and with using spectrogram is %71,3. In project (6), SVM was used and the best accuracy is %48.11.

The most common used techniques to detect emotions from voice are extracted audio features as MFCC, Short time fourier transform (STFT).

In project (1), Ramses (Recognition and Monitoring System of the Environment of Schistocerca) database was used and it has 7 classes that is different from our dataset. In project (2), there are 4 classes. They extracted features which are pitch, intensity, first four formants, bandwidths and standard deviation. In project (3), SAVEE (Surrey Audio-Visual Expressed Emotion) dataset was used. It has 7 different emotions. In project (4), Berlin Emotional Database (BES) was used. It has 5 classes. In project (5), University of Southern California's Interactive Emotional Motion Capture (USC-IEMOCAP) database was used. In project (6), RAVDESS (The Ryerson Audio-Visual Database of Emotional Speech and Song) dataset was used which is same with us.

3. Methodology

In this work, we used the speech part of RAVDESS * dataset. It contains 1440 audio files of recorded by 24 different professional actors (12 male, 12 female) and each sample is approximately 3 seconds. All samples except neutral have been recorded according to two levels of emotional intensity, normal and strong. Because, neutral expression cannot expressed with intensity. Dataset includes two similar emotions, calm and neutral. In many researches, neutral expression is defined as severe lack of emotion. It is also known as poker face. But calm expression can be thought as having more positive worth contrary of neutral. There are two statements, ”Kids are talking by the door.” and ”Dogs are sitting by the door.”. Each recording is in North American English. Dataset contains 8 different labeled emotions which are calm, happy, sad, angry, fearful, surprise, disgust, and neutral. Neutral class has 96 samples and each of other classes have 192 samples. [7]

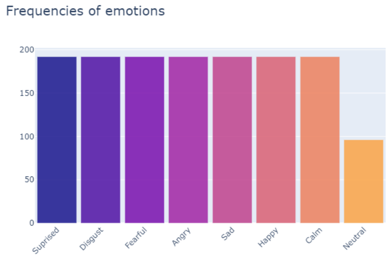

Figure 1. Bar graph of dataset about frequencies of classes

We used CNN, LSTM, Random Forest, Decision Tree, Gradient Boost, Catboost and XGBoost algorithms in this work. In section 4, results of these methods are told. We used accuracy, loss function and confusion matrices as evaluation metric.

3.1. CNN

A convolutional neural network consists of convolution layers, activation function, pooling layers and fully-connected layer. CNN is commonly used in image processing. In this work, we used CNN with MFCC features and with spectrogram and amplitude images of audios as representation of RAVDESS dataset. Especially, we expected good results in images as inputs.

Convolution layer :

A filter is applied to image. By results of these applied filters, it can detect patterns and edges and feature map is achieved. It has different types like Conv1D, Conv2D, etc.

Activation function :

Activation functions provide to transform input to non-linear form. Therefore, neural network can learn more complicated problems. If activation layer does not applied, model does not learn non-linear problems.

Pooling layer :

Pooling layer reduces the size of image and number of weights. There are many pooling processes like max pooling, min pooling, average pooling. It applies given sized filter to images and splits the image to small cells. Then, it detects cells according to chosen process and creates a new layer.

Fully-Connected Layer :

Fully-connected layer binds every neuron in current layer with previous layer.

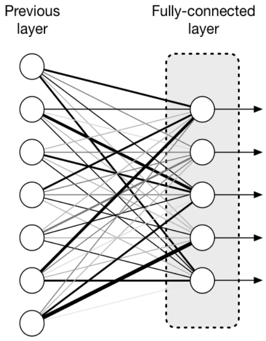

Figure 2. Visualization of fully-connected layer (taken from https://mc.ai/fully-connected-layer-with-dynamic-input-shape/)

3.2. RNN

A recurrent neural network can be thought of consecutive same networks. The input of a layer at time t is the output of the time at t-1. RNN has a memory that stores all information about the calculated. It stores calculated in memory. This process distinguishes RNNs from feedforwards. In RNN architecture, all operations are related to each other. This is called long-term dependency. These long-term dependencies have caused greater problems. Therefore, LSTM architectures have been developed.

3.3. LSTM

LSTM is an abbreviation of long short-term memory. LSTM is used especially in speech recognition area. It is a special kind of recurrent neural networks that could learn long-term dependencies. It was introduced by Sepp Hochreiter and Jürgen Schmidhuber in 1997. It was developed to avoid exploding and vanishing gradient problems [1] that can

[1] In backpropagation, when updating weights, loss value decreases gradually but gradient can approach to zero (”vanish”) or an error may occur that causes accumulating gradients and gradient can approach to infinity (”explode”).

be occur when training RNNs. Brands use LSTM as one of the fundamental components in their products. For example, Apple's Siri, Amazon's Alexa or Google Translate.

A LSTM unit is comprised of gates which are input gate, forget gate and output gate. In one unit, there are four layers that interact with each other. Following formulas represent a LSTM unit with forget gate :

$$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f)$$
$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i)$$
$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o)$$
$$c_t = f_t * c_{t-1} + i_t * c'_{t-1}$$
$$h_t = o_t * \sigma_h(c_t)$$

where $c'_{t-1} = \sigma_c(W_c x_t + U_c h_{t-1} + b_c)$ and forget gate f, input gate i and output gate o. x_t is the input vector, σ is activation function (σ_g: sigmoid, σ_h: tanh) and h_t is the output vector.

3.4. MFCC

Mel-frequency cepstral coefficients (MFCCs) are a small set of feature set which come from fourier transformation. MFCC represents characterizations of voice data and it is commonly used as feature matrix in speech recognition. Formula that use for converting the Hertz value (f) to Mel scale:

$$Mel(f) = 2595 * log(1 + f/700)$$

In this work, we used this feature as feature matrix in CNN, LSTM and Machine Learning models.

3.5. Spectrogram

Spectrogram is representation of voice with respect to time - frequency axes. In this work, we used it as CNN input.

3.6. Amplitude

Amplitude is half of vertical distance of a wave's peak point to pit point. Audios can be represented with amplitude. We visualize audios with respect to their amplitude representation and used these images as CNN input.

4. Experimental Results

In this work, we used many algorithms. Firstly, we extract features of audio like "mfcc", "chroma_stft", "chroma_cqt", "chroma_cens", "rms", "spectral_contrast", "spectral_bandwidth", "tonnetz" and "zcr". We used librosa library when extracting these features. Chroma feature or chromagram is a visualization of a feature vector that how much energy has every pitch class is in an audio signal. "chroma_stft" stands for short time fourier transformation of a waveform. "chroma_cqt" is a type of wavelet transform similar to mel scale. "chroma_cens" computes CENS (Chroma Energy Normalized). It is used for finding similarities or matchings in audio. "rms" is stands for root mean square. "spectral_contrast" returns difference of each sub-band of a spectrogram. "spectral_bandwidth" computes the band width. "tonnetz" indicates the tone of an audio. "zcr" represents zero crossing rate of an audio. We extracted the images of amplitude and spectrogram of audios to use them as input of CNN.

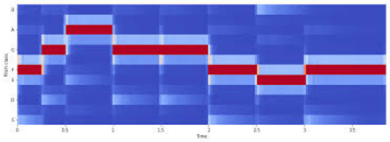

Figure 3. An example of chroma in time-pitch class axes (taken from https://musicinformationretrieval.com/chroma.html)

We used many different models. Our best models' details are explained below. We splitted our dataset into train (%70) and test (%30) sets.

4.1. Model 1 - CNN with MFCC

In this model, we used 4 Conv1D layer, -first layer has 256 units, second, third and fourth layers have 128 units and in each layer ReLU was used-, filter size 5x5, 1 Dropout layer with 0,1 as parameter (we used this dropout layer to prevent the overfitting), 5 Activation layers (the last one is Softmax and the others are ReLU), 1D MaxPooling layer, 1 Flatten layer and 1 Dense layer. Learning rate is 0,00001. Optimizer is Adam. Loss function is sparse categorical cross-entropy. Epoch is 100. Overall accuracy is 0,65.

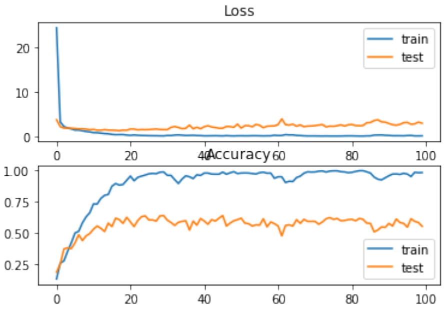

Figure 4. Loss and Accuracy graphs with respect to Train and Test

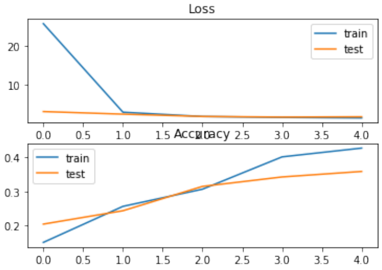

Figure 5. Same model's Loss and Accuracy graphs with respect to Train and Test with Early Stopping technique

4.2. Model 2 - CNN with amplitude images

4.2.1. Model 1

In this model, we used 2 Conv2D layer, -first layer has 64 units, second layer has 16 units and in each layer ReLU was used-, filter size 3x3, 1 Flatten layer and 1 Dense layer whose activation function is Softmax. Optimizer is Adam. Loss function is sparse categorical cross-entropy. Epoch is 10. Overall accuracy is 0,37.

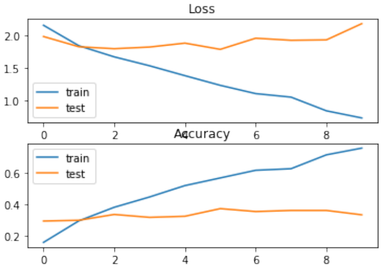

Figure 6. Loss and Accuracy graphs with respect to Train and Test

4.2.2. Model 2

In this model, we used 3 Conv2D layer, -first layer has 64 units, second layer has 16 units, third layer has 8 units and in each layer ReLU was used-, filter size 3x3, 3 2D MaxPooling layers, 1 Flatten layer and 1 Dense layer whose activation function is Softmax. Optimizer is Adam. Loss function is sparse categorical cross-entropy. Epoch is 10. Overall accuracy is 0,30.

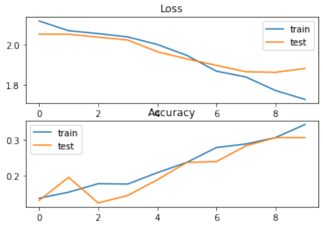

Figure 7. Loss and Accuracy graphs with respect to Train and Test

Same model with 50 epoch, accuracy is 0,36.

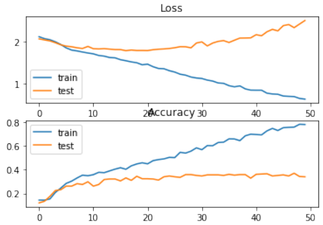

Figure 8. Loss and Accuracy graphs with respect to Train and Test in 50 epoch

4.3. Model 3 - CNN with spectrogram

In this model, we used 2 Conv2D layer, -first layer has 64 units, second layer has 16 units and in each layer ReLU was used-, filter size 3x3, 4 Dense layers whose activation function is ReLU. Last Dense layer's activation function is Softmax. Optimizer is Adam. Loss function is sparse categorical cross-entropy. Epoch is 20. The accuracy is 0,48.

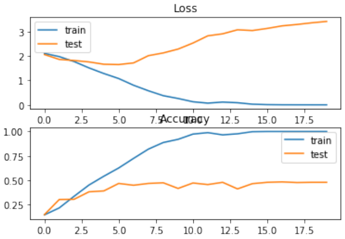

Figure 9. Loss and Accuracy graphs with respect to Train and Test

4.4. Model 4 - LSTM with MFCC

In this model, we used 4 LSTM layers and each of them has 50 units. In first 3 layers, return sequence parameter is True. Optimizer is Adam. Loss function is sparse categorical cross-entropy. Epoch is 100. Overall accuracy is 0,36.

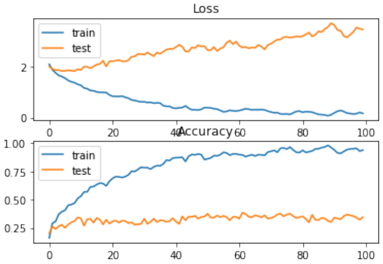

Figure 10. Loss and Accuracy graphs with respect to Train and Test

4.5. Machine Learning Models

In following methods, we used mean values of audio features we extracted. (As a reminder: 0 = neutral, 1 = calm, 2 = happy, 3 = sad, 4 = angry, 5 = fearful, 6 = disgusted, 7 = surprised)

4.5.1. Random Forest

In this model, we used Random forest as classifier. Number of estimators is 1000. Model could not learn Neutral class. Accuracy is 0,35. Model confused calm class with sad class. Calm class' accuracy is 0,62. Model could not learn Happy class. The accuracy is 0,33. Sad class' accuracy is 0,32 because model confused it with all the other classes uniformly. Angry class' accuracy is 0,48. Model confused Angry class with fearful. Fearful class' accuracy is 0,38. Model confused fearful class with mostly happy. Disgusted class' accuracy is 0,36. Surprised class' accuracy is 0,48. In conclusion, model's accuracy is 0,43.

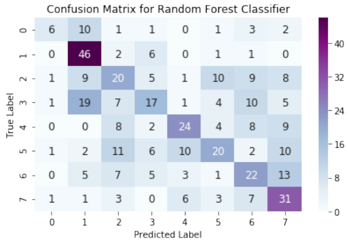

Figure 11. 0 = neutral, 1 = calm, 2 = happy, 3 = sad, 4 = angry, 5 = fearful, 6 = disgusted, 7 = surprised

4.5.2. Decision Tree

In this model, neutral's accuracy is 0,24. Calm is 0,40, Happy is 0,31, Sad is 0,28, Angry is 0,44, Fearful is 0,38, Disgusted is 0,34, Surprised is 0,37. In conclusion, neutral and calm classes confused with each other because these classes are similar. Neutral is confused with calm, calm with happy, happy with disgusted and fearful, disgusted with surprised. Sad and surprised classes are confused with other classes uniformly.

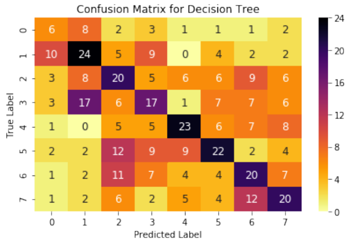

Figure 12. 0 = neutral, 1 = calm, 2 = happy, 3 = sad, 4 = angry, 5 = fearful, 6 = disgusted, 7 = surprised

4.5.3. Catboost

In this model, neutral class' accuracy is 0,26, calm is 0,60, happy is 0,34, sad is 0,34, angry is 0,50, fearful is 0,33, disgusted is 0,37 and surprised is 0,47. Overall accuracy is 0,42. Neutral class is confused with happy, calm with sad, angry with fearful, fearful with happy, surprised with fearful.

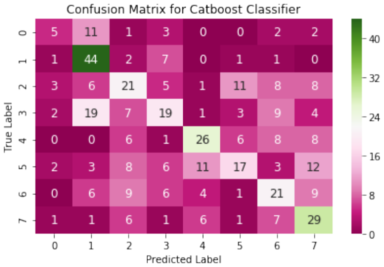

Figure 13. 0 = neutral, 1 = calm, 2 = happy, 3 = sad, 4 = angry, 5 = fearful, 6 = disgusted, 7 = surprised

4.5.4. Gradient Boost

In this model, neutral class' accuracy is 0,07, calm is 0,43, happy is 0,29, sad is 0,23, angry is 0,45, fearful is 0,35, disgusted is 0,28 and surprised is 0,42. Overall accuracy is 0,34. Neutral class is confused with calm, this is the worst model for classifying the neutral class. Calm is confused with happy and neutral. Happy, surprised and sad classes' prediction distributions are uniform.

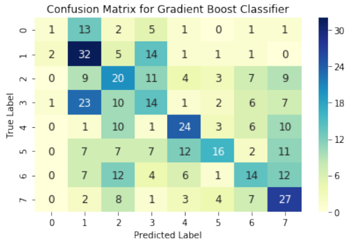

Figure 14. 0 = neutral, 1 = calm, 2 = happy, 3 = sad, 4 = angry, 5 = fearful, 6 = disgusted, 7 = surprised

4.5.5. XGBoost

In this model, neutral class' accuracy is 0,32, calm is 0,52, happy is 0,37, sad is 0,22, angry is 0,52, fearful is 0,37, disgusted is 0,32 and surprised is 0,48. Overall accuracy is 0,40. Neutral class is confused with calm and happy. Calm is confused with happy. Model confused happy with angry, surprised with fearful and disgusted.

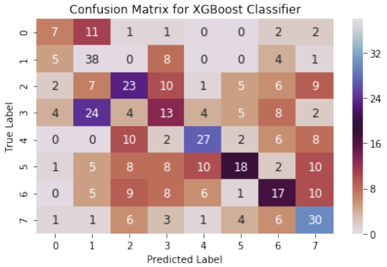

Figure 15. 0 = neutral, 1 = calm, 2 = happy, 3 = sad, 4 = angry, 5 = fearful, 6 = disgusted, 7 = surprised

5. Conclusion

In this work, we extracted audio features to use them as inputs of methods. We used mainly the CNN algorithm. We tested our CNN model with both amplitude, spectrogram and MFCC feature. we resized both amplitude and spectrogram images. We reduced the size of original images by %80 and turned into grayscale. We got the best result in CNN with MFCC feature. Accuracy, confusion matrix and loss function are used as evaluation metric. (Accuracy with respect to f1-score for machine learning methods.) According to researches, neutral and calm emotions are most confusable classes. Our results indicates this situation too. In general, all models classify correctly distinct emotions like angry and surprised but mostly sad and neutral classes were confused with calm. Results confirm this. Due to we worked with audio, besides MFCC feature we able to used different visual representations of audio. We tested the best model with own voices as validation set and results are proportional with results of training. The distinct classes such as angry, happy and disgusted were predicted correctly. But neutral class confused with other classes.

In addition to them, especially MFCC feature with CNN gave the most accurate result. Random forest classifier is the best model out of machine learning classifiers.

5.1. Future Work

In our future work, due to data we use is too clean and number of samples are not enough, we want to apply data augmentation methods such as noise injection (adding some random values into feature matrix), shifting audio time, changing pitch, changing speed of the time series to ensure that data can keep up with real-life problems. In addition, in this work, we used only the models we created. In the future, to process a big amount of image data, we think that using deep neural networks like AlexNet, VGGNet, ResNet or GoogLeNet can improve our accuracies. Also, improving LSTM would gave better results. Moreover, in concern with this project, we saw that using Hidden Markov Model (HMM), Gaussian Mixture Model (GMM) and Support Vector Machines (SVM) commonly in researches. The use of these methods will increase the accuracy of our results.

References

[1] A. Nogueiras, A. Moreno, A. Bonafonte, and José B. Mariño, ”Speech Emotion Recognition Using Hidden Markov Models”, 2001.

[2] F. Noroozi, N. Akrami, G. Anbarjafari, ”Speech-based Emotion Recognition and Next Reaction Prediction”, 2017.

[3] M. S. Sinith, E. Aswathi, T. M. Deepa, C. P. Shameema, Shiny Rajan, ”Emotion recognition from audio signals using Support Vector Machine”, 2015.

[4] Patel, P., Chaudhari, A., Pund, M., Deshmukh, D., ”Speech Emotion Recognition System Using Gaussian Mixture Model and Improvement proposed via Boosted GMM”, 2017.

[5] S. Tripathi, A. Kumar, A. Ramesh, C. Singh, P. Yenigalla, ”Deep Learning based Emotion Recognition System Using Speech Features and Transcriptions”, 2019.

[6] A. Huang , M. (P.) Bao, ”Human Vocal Sentiment Analysis”, 2019.

[7] S. R. Livingstone, F. A. Russo, ”The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English”, 2018.

[8] ”Understanding LSTM Networks”, [Online], 2015. *Available:* http://colah.github.io/posts/2015-08-Understanding-LSTMs/ [Accessed: 12-Jan-2020]

[9] A. Seker, B. Diri, H. H. Balik, ”Derin Öğrenme Yöntemleri ve Uygulamaları Hakkında Bir İnceleme”, 2017.

[10] ”Chroma Feature” [Online], 2020. *Available:* https://en.wikipedia.org/wiki/Chroma_feature [Accessed: 12-Jan-2020]

[11] ”Feature extraction” [Online], 2020. *Available:* https://librosa.github.io/librosa/feature.html#spectral-features [Accessed: 12-Jan-2020]

[12] ”Notes on Music Information Retrieval” [Online], 2020. *Available:* https://musicinformationretrieval.

com/
[Accessed: 12-Jan-2020]

[13] M. Kattel, A. Nepal, A. K. Shah, D. Shrestha, ”Chroma Feature Extraction”, 2019.

[14] ”The dummy’s guide to MFCC” [Online], 2018. *Available:* https://medium.com/prathena/the-dummys-guide-to-mfcc-aceab2450fd
[Accessed: 12-Jan-2020]

[15] ”Mel Frequency Cepstral Coefficient (MFCC) tutorial” [Online], 2020. *Available:* http://practicalcryptography.com/miscellaneous/machine-learning/guide-mel-frequency-cepstral-coefficients-mfccs/
[Accessed: 12-Jan-2020].