Classifying Emotions in Speech through Machine Learning

October 2023

Nowadays, in the era of numerous scientific and technological discoveries, our pace of work has significantly increased. This leads to many health issues, such as abnormal amount of stress, exhaustion, and many others. The key to solve this problem lies in in the early identification of these health concerns and implementing preventive measures. As a programmer myself, I was interested in solving this well-being problem using the machine learning. Imagine having a tool capable of classifying our emotions solely from the sound of our voice and offering relevant advice – such a tool could be a game-changer in tackling these issues. This project tries to pursuit this ambitious goal.

In this report in «Problem formulation» section I will first describe the datasets, which were used to train the model. Then, we will compare different machine learning methods in «Methods» part. And, finally, this report will identify the most optimal model for this purpose in conclusion

Problem formulation

The primary goal of this project is to develop a machine learning system capable of classifying the emotion of the speaker within a short live audio recording. Therefore, the problem can be determined as a Classification machine learning task, and thus it belongs to supervised learning.

In the realm of machine learning, it's a well-accepted fact that the size of the dataset affects significantly the output accuracy of the trained model [1]. This phenomenon is also evident in this project. Most publicly available voice-emotions datasets have insufficient amount of samples to achieve acceptable accuracy. To avoid this problem, four different datasets were combined, where each datapoint represents one voice audio sample.

For training and testing purposes, a subset of «The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS)» dataset has been employed [2]. This dataset contains a total of 7356 distinct files, including audio only, audio-video and video only format. For this project it was decided to explicitly utilise audio only speech files, since video and songs are unrelated to this project. The files are provided by the dataset in 16-bit resolution at a 48kHz sampling rate in .wav format. The dataset contains 1440 suitable samples, recorded by 24 actors in different variants. The second dataset for this project is Toronto emotional speech set (TESS) provided by University of Toronto [3]. This dataset contains 2800 samples in total, portraying each of seven emotions.

It is also important to mention the dataset Crowd Sourced Emotional Multimodal Actors Dataset (CREMA-D) by National Library of Medicine [4], which contains 7,442 original clips from 91 actors with one to six emotions. However, the most profound impact on the machine's accuracy was observed to originate from the "Emotional Speech Database (ESD)" by Kun Zhou et al.[5], an extensive compilation encompassing approximately 17,500 diverse samples, each representing from one to five distinct emotions.

The feature extracting process from audio files is not as straightforward as from images and can be considered as nontrivial. These features can generally be categorised into

four groups: linguistic, contextual, acoustic, and hybrid [6]. For the purposes of this project, the acoustic group of features is utilised, since it is considered to be the most effective for emotion detection [6]. This group comprises different features, including voice quality features, pitch, loudness and spectral features. Extraction of these features from the audio samples will be carried out using the Python package ‘librosa’ [7]. Specifically, the project will prioritize the use of Mel-frequency cepstrum (MFCC) with 50 coefficients, MEL Spectrogram Frequency for 128 Mel bands, tonal centroid features (tonnetz) with 6 different tonal features, spectral contrast for 7 spectral bands and a chromagram from a waveform producing 12 features. There are at total 203 different float-value continuous features

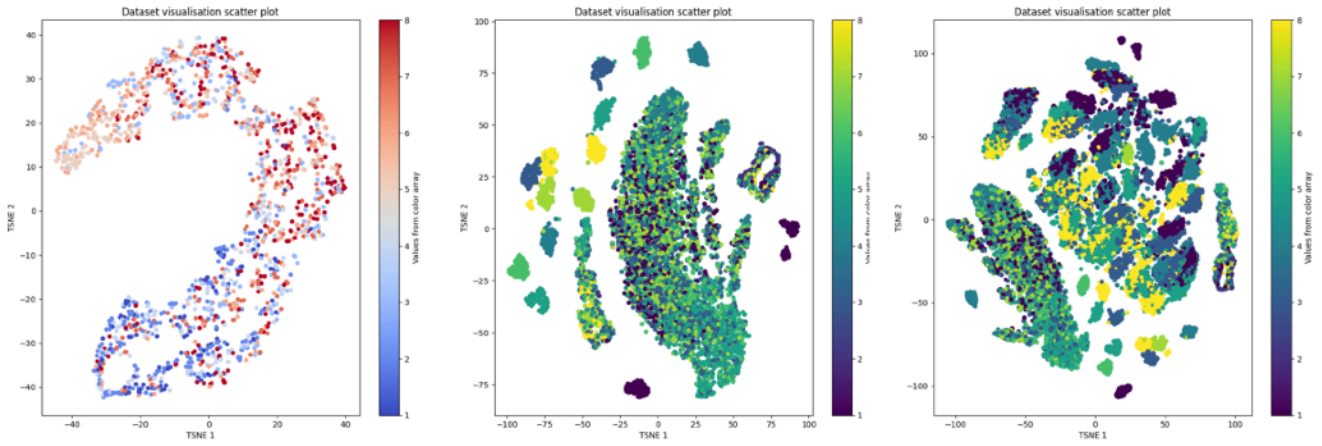
As can be seen from the dataset, the labels for this project correspond to numerical codes that represent different emotions. That is, 1 represents neutral, 2 — calm, 3 — happy, 4 — sad, 5 — angry, 6 — fearful, 7 — disgust and 8 — surprised emotion. Thus, the labels are categorical by the type.

Data preprocessing and feature selection

Since the final dataset was combined from four distinct sources containing various numbers of emotions, certain data preprocessing steps were needed to optimise the training process. Firstly, a Python function was developed for each dataset to extract the labels from the filenames and subsequently rename the files in the format of "label_individual number.wav." Furthermore, these files were relocated to the primary sample folder, streamlining the iteration process for future use. The cumulative dataset encompasses a total of 29,182 datapoints. The label numbers were changed to start from zero, thus improving the performance of the models.

As previously mentioned, it was decided to extract 203 different features from the sample. These selected features align closely with the project's objectives, as they comprehensively encompass key facets of speech, including various frequency components, tonal attributes, and contrast characteristics.

This set of features was also tested visually using scatter plots. The project employed the t-SNE algorithm to downsize the amount of features to two, representing the x and y coordinate. The plot provides also the visual comparison of different dataset sizes



1440 samples, 20 features	~12000 samples, 203 features	29182 samples, 203 features

TABLE 1

As can be seen from the Table 1, with bigger amount of features and samples, more clusters are shown, thus justifying the decision

Methods

As evident from Table 1, there is no linear relation between features and labels, so the decision was made to use some advanced machine learning methods for this project. One such method, chosen as the foundation for the initial model, is the Multi-layer Perceptron (MLP)

Multi-layer perceptron (MLP) model is widely used in machine learning due to its versatility, excelling in various tasks, including classification and regression. The main idea of working principle behind MLP is using multiple hidden layers of neurons with different activation functions between input and output layers to achieve a non-linear mapping [8]. This model accomplishes well speech recognition, so it is an obvious choice for the first model

The MLP model is provided by python «sklearn» package. For the activation function package propose the usage of Relu (rectified linear unit) function, that can be described as follows: $Relu(x) = max(0,x)$.Therefore, only positive neurons values are proposed to the next layers. Relu function is widely used among machine learning systems as it fits different problems well, and that is why it is also used in the project. With the Relu function as an activation function, the model also optimises the logistic loss function using LBFGS or stochastic gradient descent. The logistic loss function is valuable for emotion classification, as it allows the model not only to predict the most likely emotion but also to quantify its confidence in that prediction. It is also well-behaved and efficient to optimise, making the training process of the MLP model smoother and more stable. This can help in faster convergence and better model generalisation. In this particular model, the more general categorical crossentropy loss is used, that generalises the logistic loss for multi-label classification.

For the second model, the Convolutional Neural Network (CNN) has been chosen. This decision is evident due to the more complex structure of the CNN model. While the MLP model utilise only dense layers (where every neuron is connected to all neurons of the next layer), the CNN employs convolutional layers. A convolution layer applies a set of filters to input data, sliding them across the input to extract local patterns and create feature maps that capture hierarchical representations of the data [9]. Therefore, the convolution layer can determine different patterns within data, making it suitable for various classification tasks, including image detection and audio recognition That is the reason behind utilising the CNN as the second model.

The constructed model has a straightforward structure and comprises different layers, such as convolutional layer as input layer, Dense layer, max pooling, flatten and, finally, dense output layer with eight neurons. The output layer employs the softmax activation function, facilitating multi-class classification, while the other layers employ the ReLU function. To compute loss during both the training and testing phases, the sparse categorical cross-entropy loss function is utilised for the same reasons as in the MLP model.

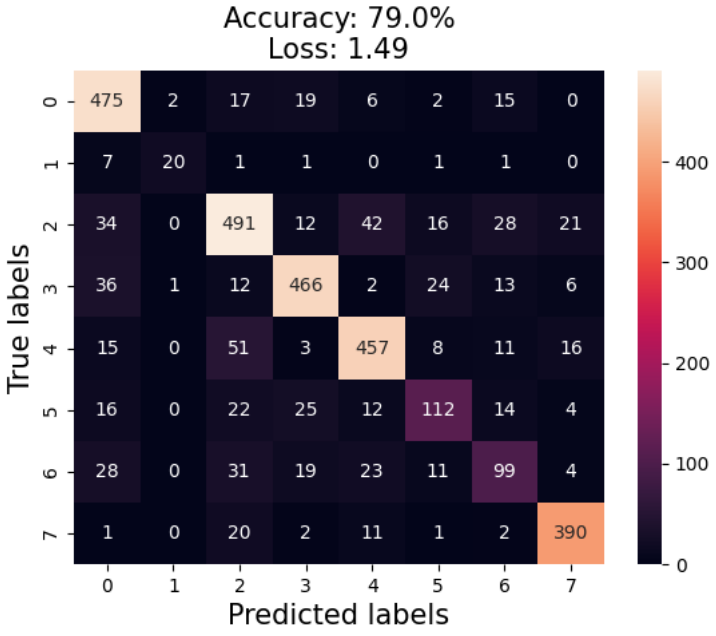
Model validation

The model is validated using standard technology by running against the validation datapoints and comparing predicted labels to validation labels. For splitting the datasets into training and validation sets, train_test_split function from sklearn package is used with the relation of 1:2 between them. Therefore, 67%, 19551 of the datapoints are used in training

and 33%, 9631 in validation and testing. This is standard size for the sets, as it provides fair distribution in samples. Shuffling the datapoints when splitting them into sets helps ensure that the data is randomly distributed across the two sets. This randomisation reduces the risk of any unintentional patterns or biases in the data affecting the model's performance and allows for more representative evaluation of the model's generalisation ability. After that, these 9631 datapoints are split into validation and testing sets with same proportions. That is, final sets comprises 19551, 6452 and 3179 datapoints. The accuracy of predicted values is calculated using sklearn's accuracy_score function, which return a float number between 0 and 1, representing the amount of right predictions in percents.

Result

After implementing, training and validation phases for each models, rather interesting results has been obtained. The loss function for the MLP model yielded a value of 1.18, which is significantly higher than 0.76 observed for CNN model. At the same time the achieved accuracy of MLP model is 79%, whereas for CNN model - 76.9%. These results indicate that while the MLP model produces a higher loss function value, it delivers superior accuracy. Therefore, the MLP model is decided to be better and is chosen to proceed with for further testing. Subsequent evaluation of accuracy and loss on the testing dataset reveals a performance of 79% accuracy and a loss value of 1.49.



Conclusion

To sum up, in this project two different models for emotions recognition were constructed. These models revealed that the Multi-Layer Perceptron (MLP) model consistently achieves superior results compared to the Convolutional Neural Network (CNN), particularly in terms of accuracy. However, the confusion matrix of the test set reveals certain limitations in accurately predicting specific emotions. The reason behind this lies in the limitation of the datasets. While all datasets contained neutral, happy, sad, angry and surprised emotions, only few of them included fear and disgust and merely one - calm. This can be mitigated by utilising, for example, larger and more complete datasets. A second limitation lies in neural networks hyperparameters tuning. Since the duration of the training process is large, it is possible that the optimal parameter values have not yet been identified. Therefore, further testing on high-performance computing systems and parameter fine-tuning may improve accuracy and potentially change the model selection. However, it is noteworthy that achieving an accuracy of approximately 80% in this non-trivial task can be considered a successful outcome for this project.

Appendices

The source code of the project can be found on a GitHub page here: https://github.com/Dereden399/voice-emotion-recognition

Sources

[1] P. Koshute, J. Zook, and I. McCulloh, 'Recommending Training Set Sizes for Classification', arXiv [cs.LG]. 2021.

[2] Livingstone, Steven R. and Russo, Frank A., "The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS)", PLoS ONE, vol. 13, no. 5. Zenodo, p. e0196391, Apr. 05, 2018. doi: 10.5281/zenodo.1188976.

[3] M. K. Pichora-Fuller and K. Dupuis, Toronto emotional speech set (TESS). Borealis, 2020. doi: 10.5683/SP2/E8H2MF.

[4] H. Cao, D. G. Cooper, M. K. Keutmann, R. C. Gur, A. Nenkova, and R. Verma, "CREMA-D: Crowd-Sourced Emotional Multimodal Actors Dataset," IEEE Transactions on Affective Computing, vol. 5, no. 4, pp. 377–390, 2014, doi: 10.1109/TAFFC.2014.2336244.

[5] K. Zhou, B. Sisman, R. Liu, and H. Li, Emotional Voice Conversion: Theory, Databases and ESD. 2022.

[6] A. S. Alluhaidan, O. Saidani, R. Jahangir, M. A. Nauman, and O. S. Neffati, "Speech Emotion Recognition through Hybrid Features and Convolutional Neural Network," *Applied Sciences*, vol. 13, no. 8, p. 4750, Apr. 2023, doi: 10.3390/app13084750.

[7] McFee, Brian, "librosa/librosa: 0.10.1". Zenodo, Aug. 16, 2023. doi: 10.5281/zenodo.8252662.

[8] C. Bento, "Multilayer Perceptron explained with a real-life example and python code: Sentiment Analysis," Medium, https://towardsdatascience.com/multilayer-perceptron-explained-with-a-real-life-example-and-python-code-sentiment-analysis-cb408ee93141.

[9] IBM, "What are convolutional neural networks?", https://www.ibm.com/topics/convolutional-neural-networks