miladshiraniUCB / **Emotion-Detection-in-Speech** Public

0 stars 0 forks

Star Unwatch

Code Issues Pull requests Actions Projects Wiki Security

main

miladshiraniUCB Project DONE ... 1 minute ago 39

View code

README.md

Emotion-Detection-in-Speech

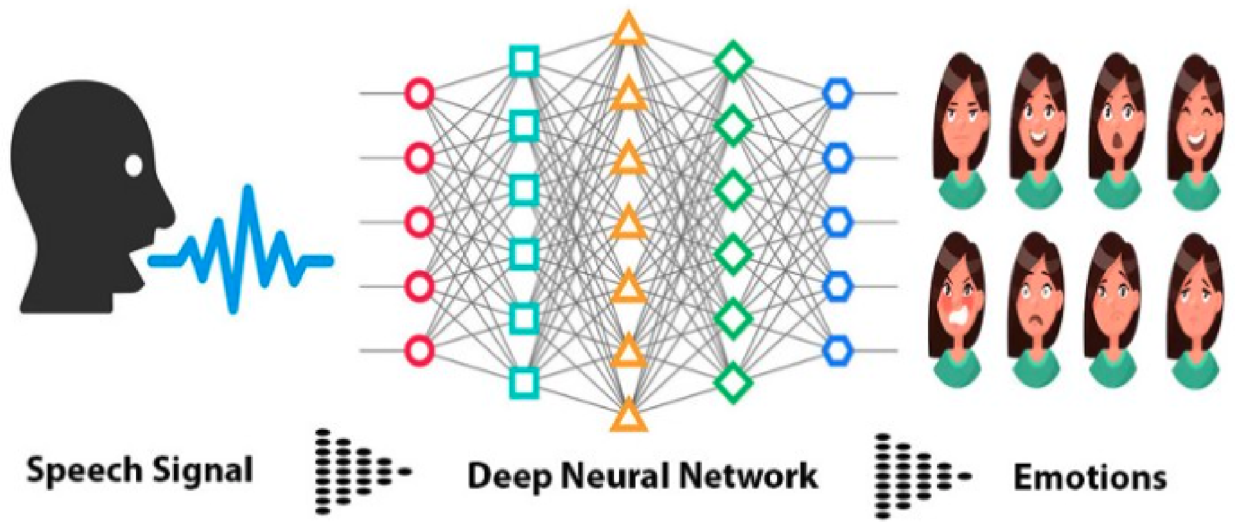


image from here

Introduction

The information in speech is conveyed through words and emotion. Depending on how one pronounces a word, we can understand different meaning. Therefore, both the word and the way it is pronounced affect our understanding of the word. As a result, it would be important that we could design a virtual assistant that not only does understand the word, but also it understands the emotion in the way the word is pronounced. this virtual assistant can be used to assist therapists to fully analyze a patient, because it is important to understand not only the meaning of the word and how they are put together to make a sentence, but also it is important to understand the emotions conveyed in a sentence by the patient. So, we want to design and introduce a model to assist therapists with analyzing the emotion of the sentences more accurately.

In this work, we are trying to introduce a machine learning model that can detect the emotion in the sentence. In order to do so, we are using the audio data provided by University of Toronto to create a neural network model to detect the emotion in a speech. In order to make a model we need to convert the audio files into numerical values. To convert the audio file to a numerical data, we use python library librosa and to denoise them we used noisereduce. Afterward we convert the denoised audio files into their corresponding spectrograms using which we can train a Convolutional neural network.

Project Structure

The arrangement of this work is as follows

1. The folder `Toronto-Data` contains the audio files we used in this work.
2. The folder `Notebook` contains the following notebooks:
   - The Exploratory Data Analysis (EDA), Train-Test split and conversion of audio files to spectrograms are done in the notebook `EDA-and-Data-Visualization.ipynb` .
   - The baseline modeling is performed in `Modeling-LR-XGB-LGBM-TREE.ipynb` . In this notebook, we used the numerical values obtained by converting the audio files to train several categorical classification models which are Logistic Regression, Decision Tree, and ensemble models such as Random Forest, XGBoost and LightGBM.
   - In Modeling-CNN-and-Transfer-Learning.ipynb we presented some neural network models in which we used 2D convolutional layers as well as `EfficientNetB3` and `EfficientNetB7` .
3. The folder `Train-Test-Split` contains the training and testing dataframes.

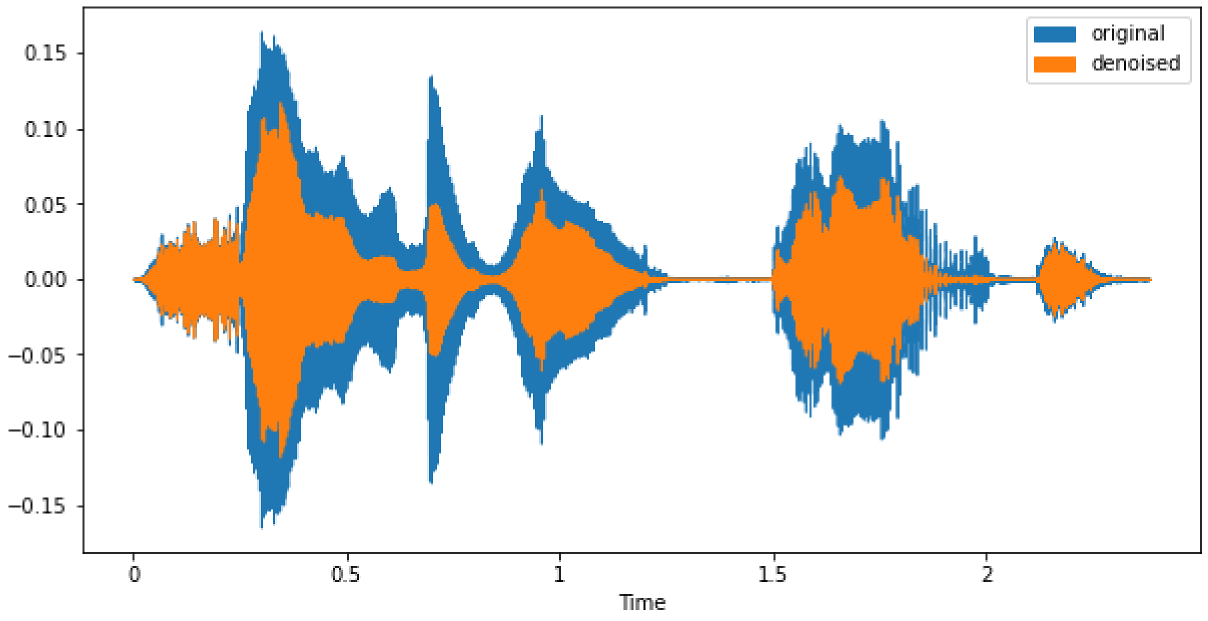
4. The folder `mel_spectrogram` contains the spectrograms of the audio files.

Repository Structure

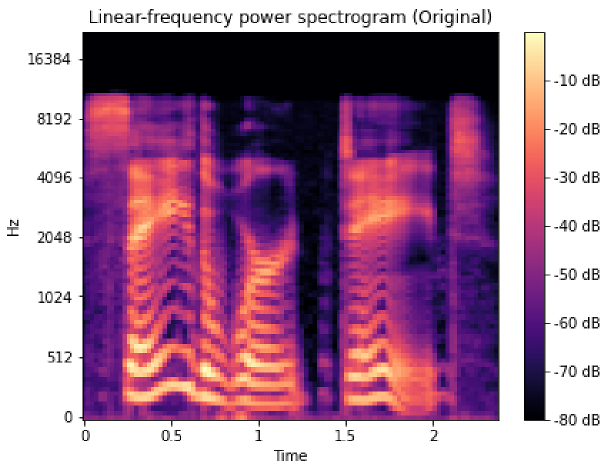
```
├── NImages: Images for README
├── Notebook: Modeling Notebooks
├── PDFs: PDF files of the notebooks and Presentation
├── Toronto-Data : data used for modeling
├── Train-Test-Split: dataframes for creating train and test sets
├── mel_spectrogram: mel spectrograms of the audio files to be used for
modeling
├── README.md : project information and repository structure
```

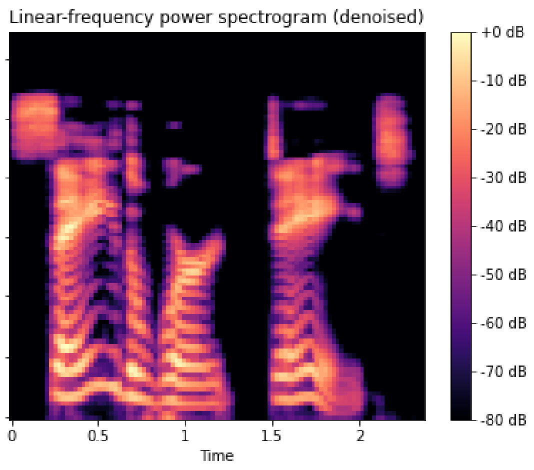
Some Visualizations

In the following figure we showed the effect of denoising of an original audio file



and the corresponding spectrogram for the same audio file for the original and denoised cases are shown below



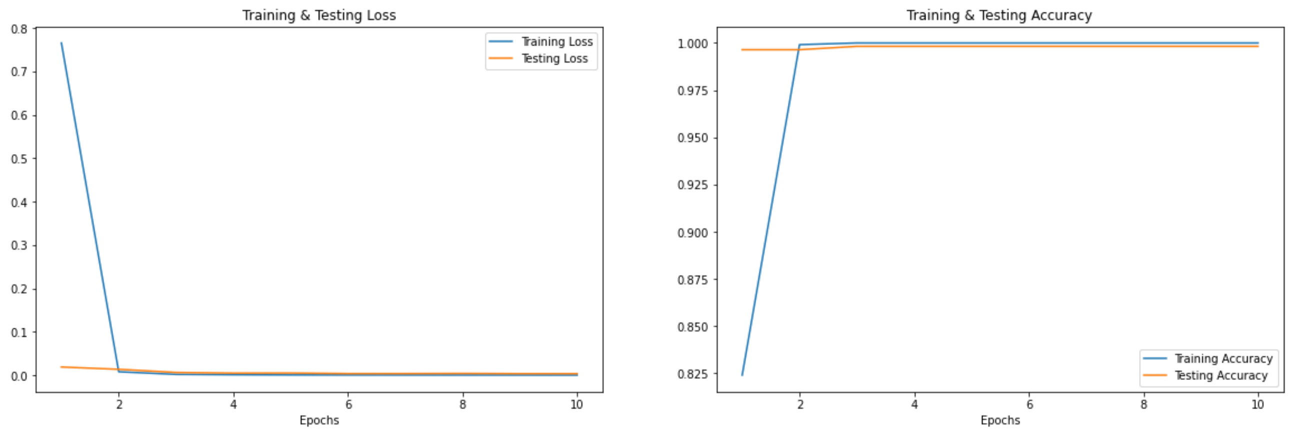


Results and Final Model.

All the neural networks we introduced in the notebook Modeling-CNN-and-Transfer-Learning.ipynb performed well with the test accuracies around 99% (the transfer learning model `EfficientNetB7` has the lowest test accuracy which is about 0.95% after 35 epochs). However, we will choose the first neural network model as the final model of this work since this model has the simplest structure and its train and test losses converged after only 2 epochs. The summary of this model is shown below

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_28 (Conv2D)          (None, 148, 148, 16)      448

 max_pooling2d_28 (MaxPoolin  (None, 74, 74, 16)       0
 g2D)

 flatten_10 (Flatten)        (None, 87616)             0

 dense_36 (Dense)            (None, 64)                5607488

 dense_37 (Dense)            (None, 7)                 455

=================================================================
Total params: 5,608,391
Trainable params: 5,608,391
Non-trainable params: 0
_________________________________________________________________
```

the results of this model is shown below



Recommendations to Improve the Model

For the future work, we would recommend using more data to train the model with and also we would recommend exploring neural network models in which LSTM layers are used and train them with the numerical values of audio files without converting them to mel-spectrograms. Moreover, we would suggest using MFCCs (Mel Frequency Cepstral Coefficients) to train machine learning models.

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Languages

● **Jupyter Notebook** 100.0%