

Capstone Project

Speech Emotion Recognition

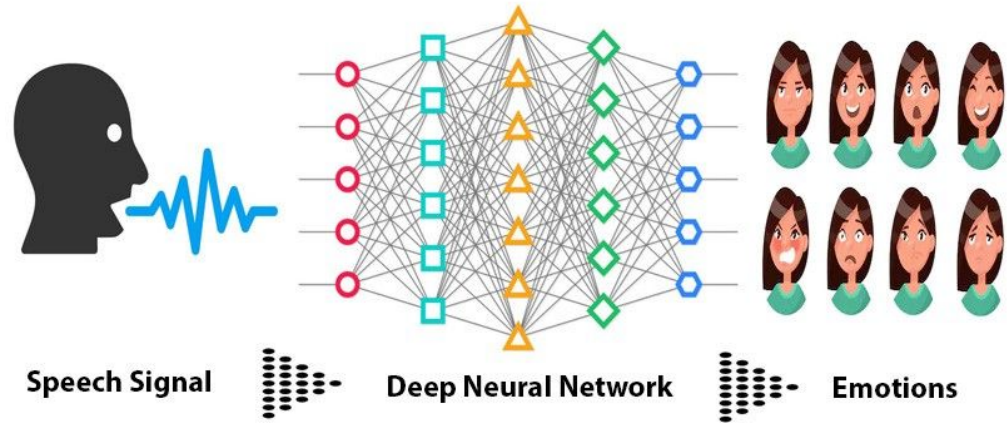


Created By : Sakshi Dhyani

Content

- ❑ **Project Details**
- ❑ **Data Description**
- ❑ **Exploratory Data Analysis**
- ❑ **Model Implementation - SVC (Support Vector Classifier)**
- ❑ **Model Implementation - MLP (Multi-layer Perceptron)**
- ❑ **Model Implementation - CNN (Convolutional Neural Network)**
- ❑ **Deployed app details**

Project Details



Speech Emotion Recognition (SER) is the way of identifying the emotions behind the speech. The tone, pitch and other factors related to audio help in detecting the emotions of humans while speaking. Speech Emotion Recognition has diversified applications in the field of security, medicine, entertainment and education. Features can be extracted from the audio files. Those features will be used to detect emotions.

Data Description

TESS (Toronto Emotional Speech Set) :

There are 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, pleasant surprise, sadness, and neutral). There are 2800 data points (audio files) in total.

RAVDESS Data Set:

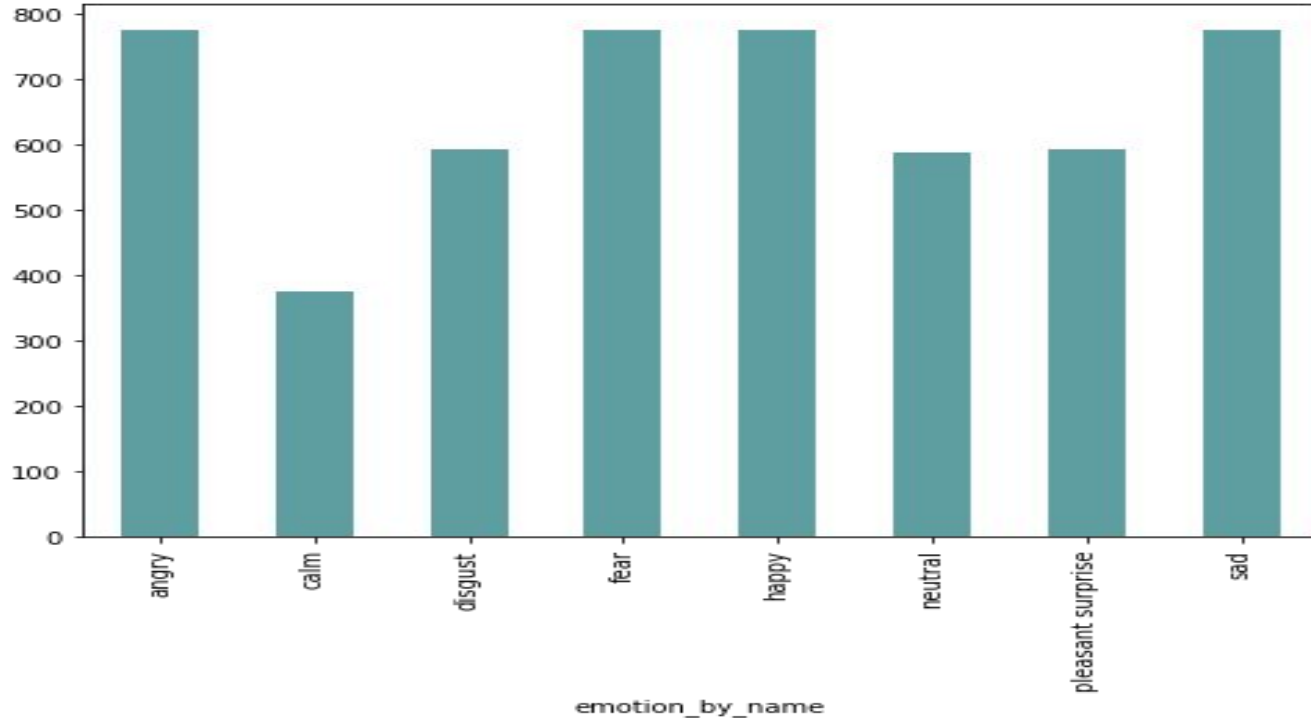
Speech data contains 1440 files: 60 trials per actor x 24 actors = 1440. The RAVDESS contains 24 professional actors (12 female, 12 male), vocalizing two lexically-matched statements in a neutral North American accent. Speech emotions includes calm, happy, sad, angry, fearful, surprise, and disgust expressions. Song data contains 1012 audio files.

Total audio files: 5252 (1012+1440+2800)

MFCC (Mel-frequency cepstral coefficients) are set of features which describes the spectral envelope. Features were extracted using MFCC.

Exploratory Data Analysis

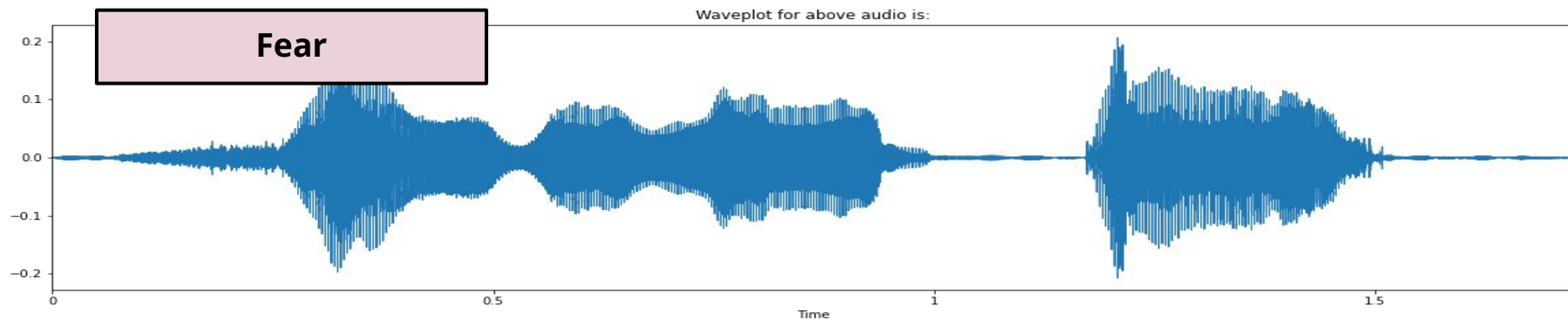
Count of audio file labels which represents emotions



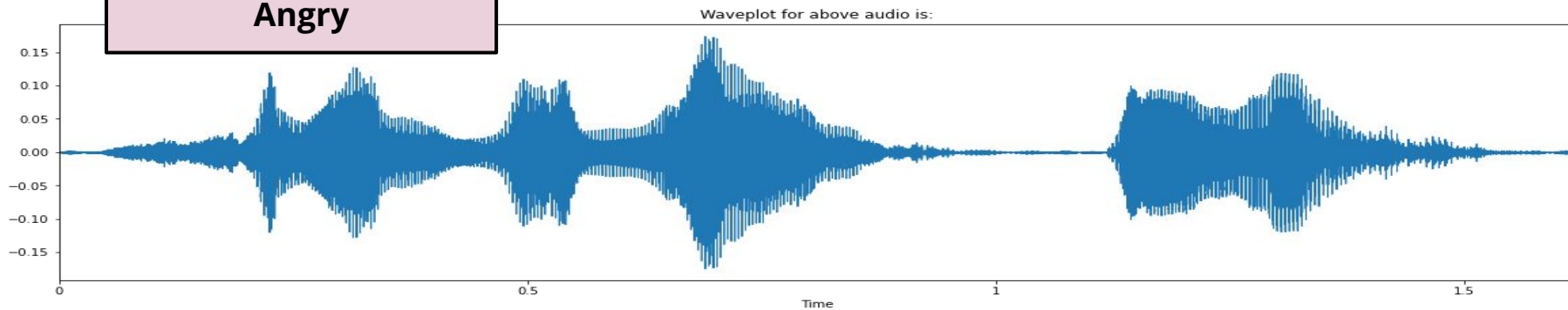
Exploratory Data Analysis

Waveform for different emotions

Fear

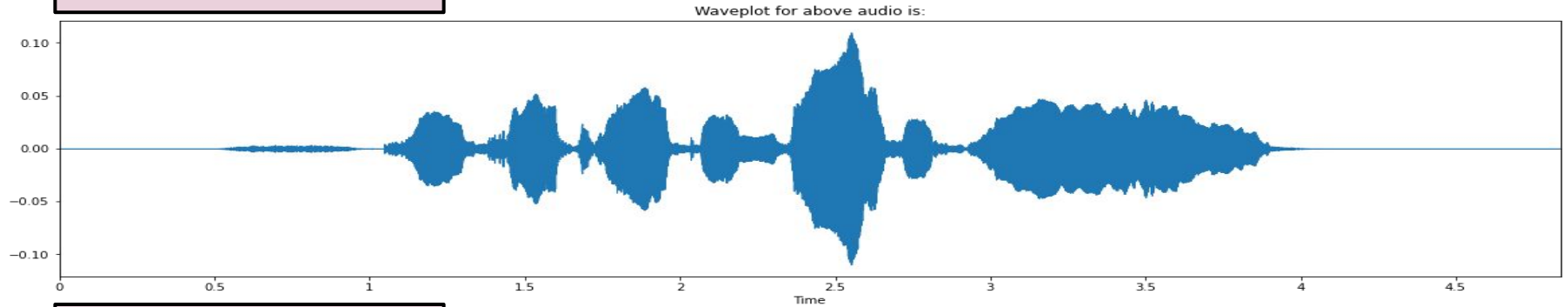


Angry

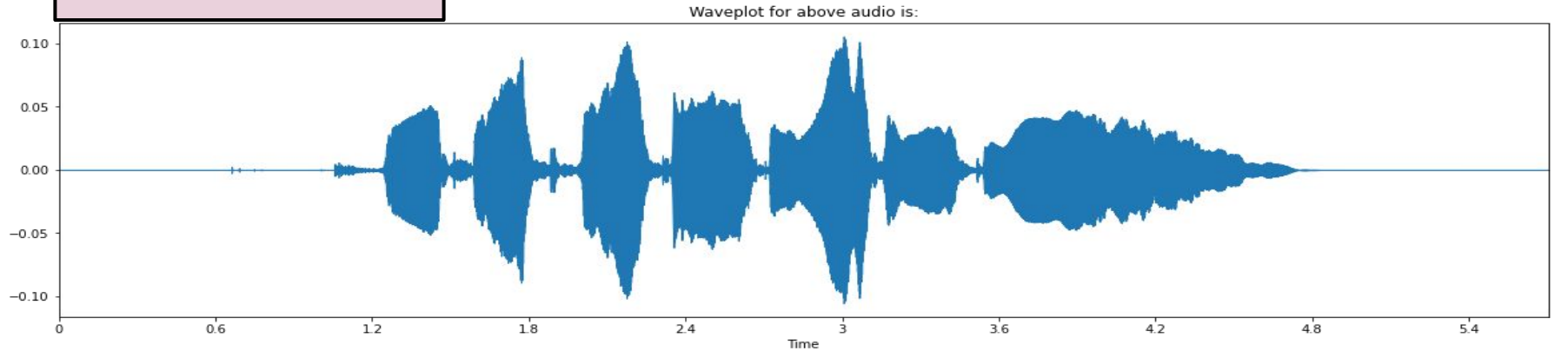


Waveform for different emotions

Happy

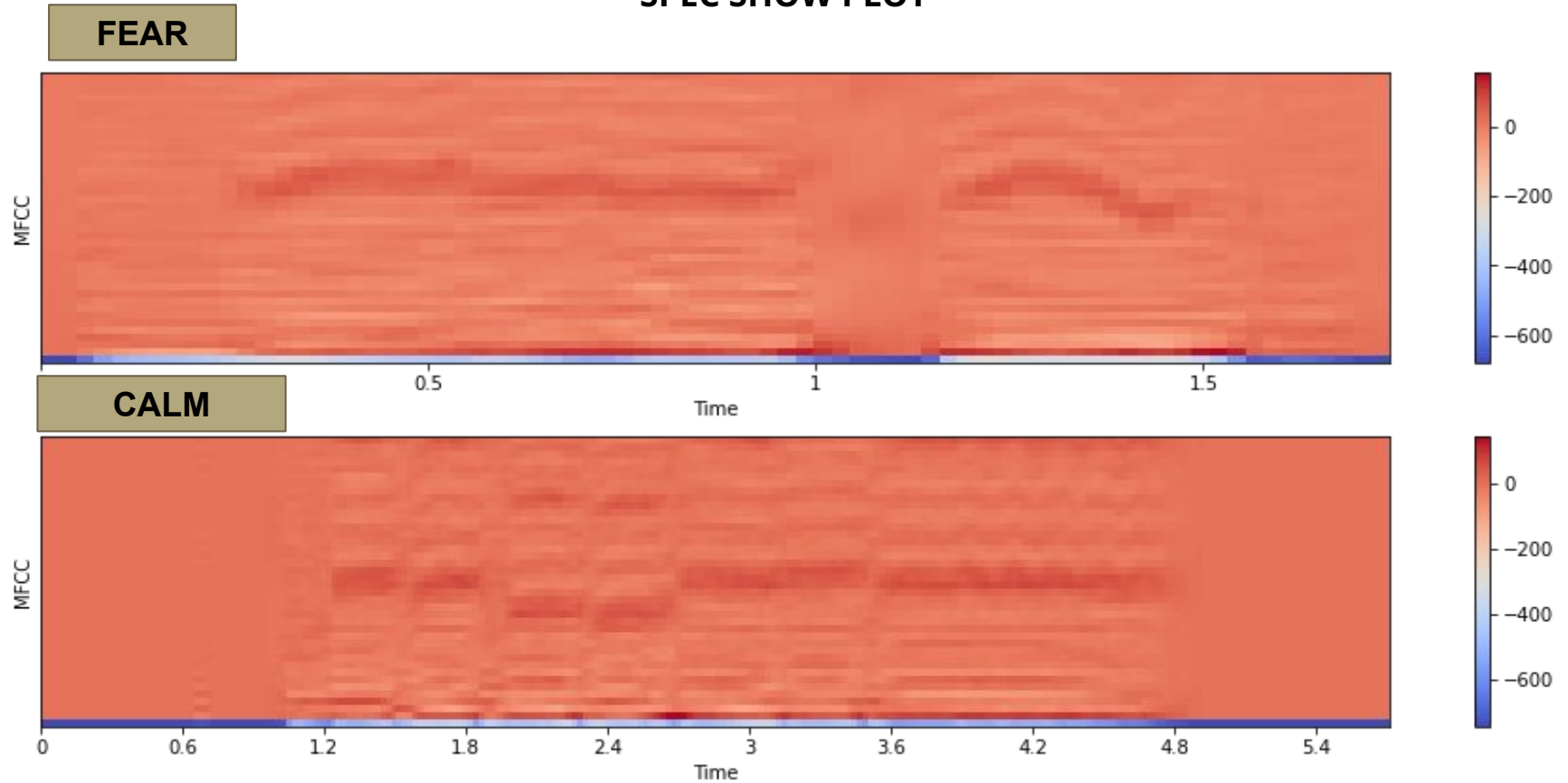


Calm



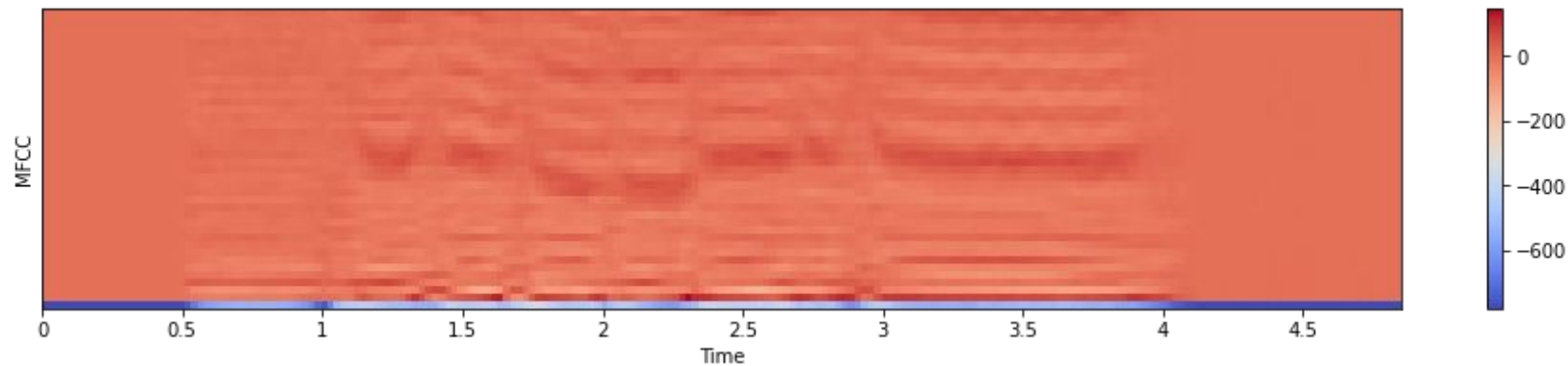
Exploratory Data Analysis

SPEC SHOW PLOT

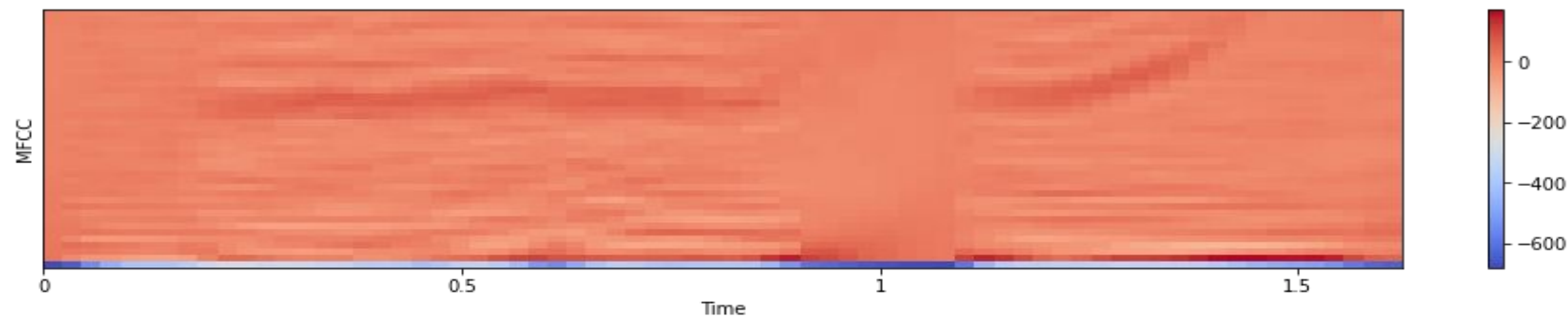


SPEC SHOW PLOT

HAPPY



ANGRY



Models Used

SVC (Support Vector classifier)

SVM is supervised machine learning algorithm used for classification, regression and outlier detection as well. SVM has classifications: SVC, NuSVC, LinearSVC. The implementation of SVC is based on libsvm. It provides best fit hyperplane to categorize the data. The multiclass support is handled according to a one-vs-one scheme. In this case, RBF kernel was used as parameter.

Training data accuracy Score: 0.882965219598883

Test data accuracy Score: 0.8088347296268088

Models Used

MLP (Multilayer Perceptron)

MLP Classifier stands for Multi-layer Perceptron classifier. MLP is a feedforward artificial neural network model that maps input data sets to a set of appropriate outputs. An MLP consists of multiple layers and each layer is fully connected to the following one. The nodes of the layers are neurons with nonlinear activation functions, except for the nodes of the input layer. Between the input and the output layer there may be one or more nonlinear hidden layers.

Training data accuracy Score: 1.000

Test data accuracy Score: 0.864

Models Used

CNN (Convolutional Neural Network)

Convolutional neural networks are distinguished from other neural networks by their superior performance with image, speech, or audio signal inputs. They have three main types of layers: Convolutional layer, pooling layer and fully-connected layer. The convolutional layer is the first layer of a convolutional network. While convolutional layers can be followed by additional convolutional layers or pooling layers, the fully-connected layer is the final layer. The convolutional layer is the core building block of a CNN, and it is where the majority of computation occurs.

CNN (Convolutional Neural Network)

Model: "sequential"

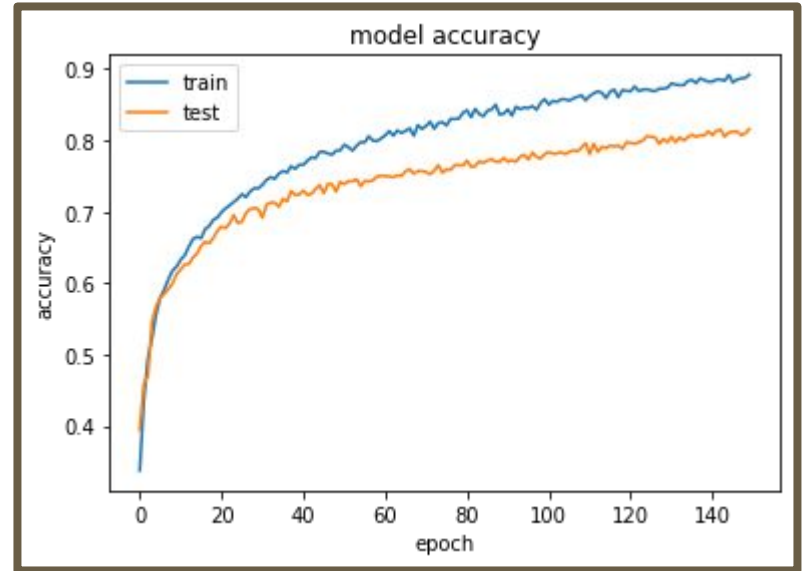
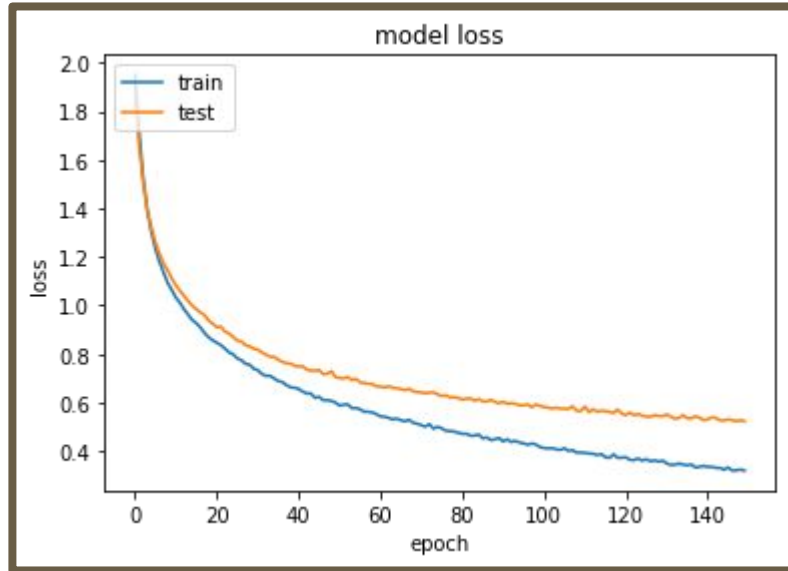
Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 40, 64)	384
activation (Activation)	(None, 40, 64)	0
dropout (Dropout)	(None, 40, 64)	0
max_pooling1d (MaxPooling1D)	(None, 10, 64)	0
conv1d_1 (Conv1D)	(None, 10, 128)	41088
activation_1 (Activation)	(None, 10, 128)	0
dropout_1 (Dropout)	(None, 10, 128)	0
max_pooling1d_1 (MaxPooling1D)	(None, 2, 128)	0
conv1d_2 (Conv1D)	(None, 2, 256)	164096
activation_2 (Activation)	(None, 2, 256)	0
dropout_2 (Dropout)	(None, 2, 256)	0
flatten (Flatten)	(None, 512)	0
dense (Dense)	(None, 8)	4104
activation_3 (Activation)	(None, 8)	0

=====
Total params: 209,672
Trainable params: 209,672
Non-trainable params: 0

Training Data Accuracy Score: 0.9324

Testing Data Accuracy Score: 0.821

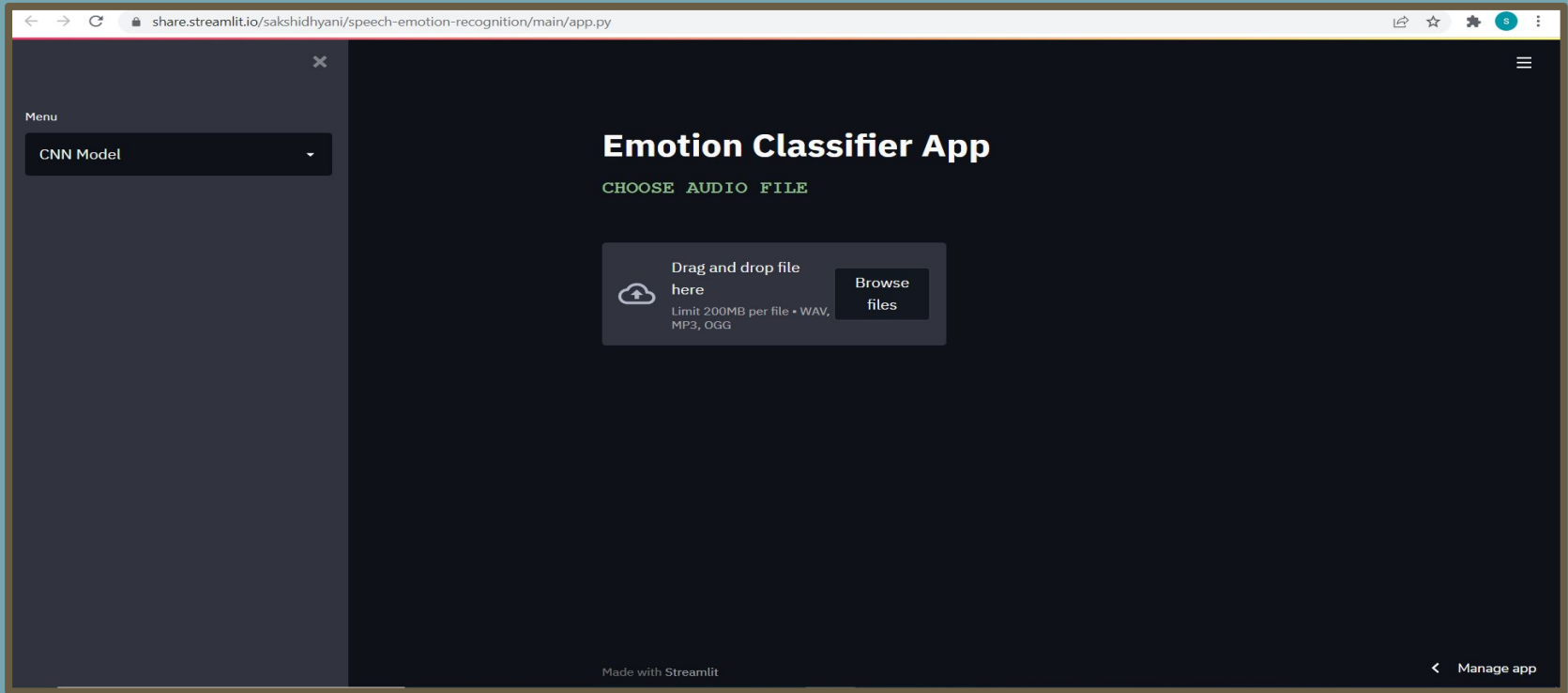
Accuracy & Loss Plots for CNN



Streamlit Web Application

Streamlit is an open-source python framework for building web apps for Machine Learning and Data Science. We can instantly develop web apps and deploy them easily using Streamlit. Streamlit allows you to write an app the same way you write a python code. Streamlit makes it seamless to work on the interactive loop of coding and viewing results in the web app.

Interface for deployed app



Deployed App to share.streamline.io:

<https://share.streamlit.io/sakshidhyani/speech-emotion-recognition/main/app.py>

Result for an audio file provided using CNN Model

share.streamlit.io/sakshidhyani/speech-emotion-recognition/main/app.py

Menu

CNN Model

Emotion Classifier App

CHOOSE AUDIO FILE

PLAY AUDIO

Drag and drop file here

Limit 200MB per file • WAV, MP3, OGG

Browse files

03-02-02-02-01-02-1... 467.1KB

0:00 / 0:04

EMOTION DETECTED

WAVE FORM

CALM

Manage app

Result for an audio file provided using MLP Model

share.streamlit.io/sakshidhyani/speech-emotion-recognition/main/app.py

Menu

MLP Model

Emotion Classifier App

CHOOSE AUDIO FILE

PLAY AUDIO

Drag and drop file here
Limit 200MB per file • WAV, MP3, OGG

Browse files

03-02-02-01-02-1... 467.1KB

WAVE FORM

EMOTION DETECTED

CALM

Manage app

CONCLUSION

SER (Speech Emotion Recognition) project helped to understand the transformation of audio, sound files into features.

MFCC extraction of features, helped in creating a proper dataset to which we can implement different models.

Performances of all the models SVC, MLP, CNN was good. Standardization of the data helped to improve accuracy score.

THANK YOU