Emotion Recognition in Speech

A Report Submitted
in Partial Fulfillment of the Requirements
for the Degree of
Bachelor of Technology
in
Computer Science & Engineering

by

Deepak Bulani(20153026)
Gaurav Agarwal(20154097)
Bonthu Harsha Vardhan Reddy(20154148)
Avichal Verma(20154085)
Amit Ranjan(20154101)

to the
COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
MOTILAL NEHRU NATIONAL INSTITUTE OF TECHNOLOGY
ALLAHABAD
PRAYAGRAJ
May, 2019

UNDERTAKING

We declare that the work presented in this report titled "*Emotion Recognition in Speech*", submitted to the Computer Science and Engineering Department, Motilal Nehru National Institute of Technology, Allahabad, for the award of the **Bachelor of Technology** degree in **Computer Science & Engineering**, is our original work. We have not plagiarized or submitted the same work for the award of any other degree. In case this undertaking is found incorrect, we accept that our degree may be unconditionally withdrawn.

May, 2019
Prayagraj

(Deepak Bulani, 20153026)

(Gaurav Agarwal, 20154097)

(Bonthu Harsha Vardhan Reddy, 20154148)

(Avichal Verma, 20154085)

(Amit Ranjan, 20154101)

'

CERTIFICATE

Certified that the work contained in the report titled "*Emotion Recognition in Speech*", by
Deepak Bulani(20153026)
Gaurav Agarwal(20154097)
Bonthu Harsha Vardhan Reddy(20154148)
Avichal Verma(20154085)
Amit Ranjan(20154101)
has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

Dr. Dinesh Singh
(Assistant Professor)
Computer Science and Engineering Dept.
M.N.N.I.T Allahabad,Prayagraj

May, 2019

Preface

We have performed emotion detection in speech for our project as human emotions serve as a backbone to all the actions and decisions we make in our life. Emotional states can motivate humans actions, and can also supplement the meaning of communication. As we know, speech is the most natural and efficient way of communication between humans. Communication between humans is actually not just what humans say, but also how they say it. People have put in efforts to develop an interface so that one can easily and effectively interact with computers.Therefore, we chose to use speech as a medium for interpretation of emotions and for further analysis and extrapolation.In this project we have focused on four integral human emotions (Happy, Sad, Anger and Neutral).Thus, we took audio signals, preprocessed it and tried to extract features out of it. Then, trained the classifier on it and designated an emotion to the input speech signal. Compressing the magnitude of the input signal without causing any harm to the power of speech signal and extracting those features from the input signal that help in identifying the speaker encompasses feature extraction . We have used Mel Frequency Cepstral Coefficents (MFCCs) for extracting features from speech. It is based on the way how humans auditory system perceives. The extracted features are then fed into various machine and deep learning classification algorithms to obtain a model for detecting emotions in speech. In the end we will analyse the results and use for standard applications.

Acknowledgements

We would like to express our sincere gratitude towards Dr. Dinesh Singh, for his patient guidance and valuable support. Without his co-operation and assistance, this project could not have been completed. From aiding us with a practical database to enhancing the quality of our project and also patiently resolving our doubts, Dinesh Sir has been the forerunner in making this project happen.
His dedicated attitude and keen interest above all his overwhelmingly friendly attitude to help his students, his timely advice, meticulous scrutiny and scientific approach have helped us to a very great extent to accomplish this task. From the start, by allowing us to choose the project of our choice to sharing his pearls of wisdom on ways to augment the worthiness of our project, we all are greatly indebted to dinesh sir.
In addition, we would like to extend out thanks to the colleagues who provided valuable insight and expertise that greatly assisted in making this research work comprehensive and precise. We would also like to thank our friends who have provided us with sources to gather information from, research and analyze the results that have been obtained after completing the project. We would also like to thank all those people who in some way or the other have helped us in developing this report .

Contents

Preface **iv**

Acknowledgements **v**

1 Introduction **1**

2 Related Work **3**

3 Proposed Work **5**

3.1 Mel Frequency Cepstral Coefficients (MFCC) 6
3.2 Linear Support Vector Machine(LSVM) 6
3.3 Random Forest Classifier 8
3.4 Multi Layer Perceptron(MLP) 9
3.5 Convolutional Neural Networks(CNN) 10
3.6 Long Short Term Memory(LSTM) 11

4 Experimental Setup and Results Analysis **12**

4.1 Dataset 12
4.1.1 Berlin's Emotional Speech Database 12
4.2 Software And Libraries Used 14
4.2.1 Anaconda 14
4.2.2 Python 3.0 14
4.2.3 Scikit-learn 14
4.2.4 Keras 15
4.2.5 NumPy 15
4.3 Models Used 16

4.4 Experiment 18
4.4.1 Preprocessing 18
4.4.2 Training 18
4.4.3 Testing 19
4.4.4 Comparison of Results 20

5 Conclusion and Future Work 26
5.1 Conclusion 26
5.2 Future Work 27

References 28

Chapter 1

Introduction

Speech is the most natural way of interaction between humans. Speech based devices are used in our daily lives and are a boon for people with speech related disabilities. Speech signals are a function of time variable which is independent and are random signals. Speech signal contains information about speech style, words, expression, accent type, emotion, sex, age, speaker's identity, the state of health of the speaker etc. Here we are more interested in extracting emotions in speech.

In applications like professional driving or flying, or anything that requires continuous monitoring of a persons emotion could use computers to encourage detection of emotions, so as to upgrade the recognition procedure and to utilize such data to help improve decisions and to support the basic leadership process. Up until this point, results emerging from machine feeling acknowledgment indicated better execution when contrasted with human feeling acknowledgment.

Feeling acknowledgment from the speaker's discourse is troublesome in light of the accompanying reasons: In separating between different feelings which specific discourse highlights are increasingly valuable isn't clear. As a result of the presence of various sentences, speakers,style of speaking,rate of talking addressing fluctuation was introduced,causing discourse highlights to legitimately get influenced. The same expression may demonstrate various feelings. Every feeling may relate to the various segments of the verbally expressed articulation. Accordingly it is hard to separate these bits of articulation. Another problematic factor is the dependence of emotional expression on the speaker and his or her culture and environment.The speaking style changes with the

culture and environment of the speaker which poses challenge for speech emotion recognition systems. It is unclear whether the recognition system will detect the emotions correctly when there are two or more types of emotions,long term and short term.
As there are six universal emotions anger, sad, happy, neutral, fear and disgusted but we have not sufficient dataset for classifying all six emotions so we have classified four emotions sad, happy, anger and neutral. The initial phase in getting machines to perceive feelings is to accumulate information from which a machine will learn. This should be possible by utilizing essential and optional inputs. The essential information could be accounts of entertainers which express various feelings by perusing the equivalent content, while the optional info could be utilizing as of now existing databases, which were created by other scientists. Some half breed approaches are likewise conceivable. The following stage would extricate highlights from gathered or obtained data. Some acoustic features are intensity, pitch, loudness. The dataset we have used in our project is Emo-DB which contains about 500 utterances spoken by actors in states of fear,happiness,anxiety,boredom,angry and disgusted as well as neutral.In order to run these different machine learning algorithms this Emo-DB dataset needs some preprocessing steps which include padding, filtering etc. And for extracting features from speech we used well known technique known as Mel Frequency Cepstral coefficients (MFCCs). Mel frequency cepstral coefficients (MFCCs) is broadly utilized in discourse acknowledgment and discourse feeling acknowledgment frameworks since acknowledgment rate of the MFCC is generally amazing. MFCC can help us achieve better robustness and frequency resolution to noise in low frequency region. Mel frequency cepstral illustrates short term power spectrum of sound.
A computer program that decides whether an sample is a positive or negative is called a classifier. A classifier is trained on hundreds of thousands of speeches to learn how to classify a new speech correctly. We have used various machine learning and deep learning classifiers in this project like Support Vector Machine(SVM), Random Forest Classifier(RF), Multi Layer Perceptron(MLP), Convolutional Neural Networks(CNN) and Long Short Term Memory(LSTM).

Chapter 2

Related Work

Before starting the project we explored the topic and went through the following research papers.They provided us a good insight into the working of MFCC for feature extraction from speech signals, preprocessing techniques for audio signals,knowledge of working of various machine learning and deep learning algorithms and training these models using our dataset.

E. Ramdinmawii, A. Mohanta and V. K. Mittal [1] This paper takes a stab at breaking down discourse and arranging it into four feeling states (Anger, Fear, Neutral, Happy) utilizing discourse signals. The examination of these feeling states got from the above highlights, in particular immediate key recurrence utilizing Zero Frequency Filtering, Formant frequencies (F1, F2, F3), overwhelming frequencies and sign vitality.
M. Ghai, S. Lal, S. Duggal and S. Manik [2] In this paper, a way to deal with feeling acknowledgment in sound signal dependent on Random Decision Forest, SVM and Gradient Boosting classifiers was displayed. The performance of the classifiers can support up essentially if appropriate features are appropriately acquired. Consequently, we presumed that consideration of vitality as an element alongside other 13 MFCC highlights prompted better appraisal of the feeling joined with the speech.It can be seen that incorporating outlines into covering fragments lead to a more progression in tests and furthermore brought about every data point having a lot more features.
Parwinder Pal Singh, Pushpa Rani [3] This research paper encompasses successful denoising of the input sample and consideration of delta energy function while extracting MFCC coefficients.A conclusion was drawn that the MFCC coefficients can be increased

as per our requirement.Addition of velocity and acceleration can be considered while extracting while extracting 39 MFCC coefficients

S. Basu, J. Chakraborty, A. Bag and M. Aftabuddin[4] As indicated by this paper distinguishing feelings of an individual is yet to have a general and complete solution. Till now, the vast majority of the specialists have worked upon fixed size speech portion for grouping of feeling, that implies on the disconnected speech. The issue emerges when speech tests are of various size, for this kind of information, sparsity of the information highlights grid causes issues. Despite the fact that standard feed-forward Multi Layer Perceptron(MLP) is an amazing asset for classification issues, very massive grid may not output ideal outcome however exploring different appropriately tuned MLP system ought to be interesting.

Chapter 3

Proposed Work

Our entire project is divided into the following parts :

1. Searching for appropriate speech database,reading audio signals and padding them to have an appropriate length
2. Extracting features from the preprocessed audio signals so that we can train various classification algorithms.We have one approach for it:
   - Mel Frequency Cepstral Coefficients (MFCC)
3. Feeding the extracted features into various machine learning and deep learning classification algorithms to obtain a model for detecting emotions in speech.Various algorithms used are:
   - Linear Support Vector Machine(LSVM)
   - Random Forest Classifier
   - Multi Layer Perceptron(MLP)
   - Convolutional Neural Networks(CNN)
   - Long Short Term Memory(LSTM)
4. Analysing the results of classification algorithms by plotting confusion matrix and calculating Accuracies and F1-score for each.

3.1 Mel Frequency Cepstral Coefficients (MFCC)

The initial phase in any automatic speech recognition system is to extract highlights for example recognize the parts of the sound sign that are useful for distinguishing the linguistically useful content and for removing the parts which shows data like background noise. The most important concern to comprehend about speech is how it is being generated.The sounds produced by a human are adjusted by vocal tract's shape. This includes tongue,teeth, jaw, etc. Different shapes are responsible for producing different sounds.In the event that we can decide the shape precisely, this should give us an exact portrayal of the phoneme being created.The vocal tract's shape manifests itself in the envelope of short time power spectrum,and MFCC's job is to represent this envelope accurately.
Prior to the introduction of MFCCs, we did not have any advanced speech recognition tool. We had Linear Prediction Cepstral Coefficient (LPCC's) which were employed in the industry for Automatic Speech Recognition. After the advent of Mel Frequency Cepstral Coefficents (MFCCs), MFCC's have become a widely famous and used as a tool in the modern industry. Their applications lies in the areas like automatic speaker and speech recognition. MFCC was founded by Davis and Mermelstein in the 1980's.

3.2 Linear Support Vector Machine(LSVM)

Data Classification is a typical task in AI. Assume we have two classes and a few information indicates each having a place one of two classes and our objective is to order the focuses to the classes. On account of support vector machines,we see an information point as a p dimensional vector, and we need to know whether these focuses can be isolated by a (p-1) dimensional hyperplane. This is known as a straight grouping. Presence of numerous hyperplanes are possible.So that hyperplane is picked which amplifies the separation from it to closest information point on each side.If there is presence of such a hyperplane it is called most extreme edge hyperplane and comparing classifier is called greatest edge classifier.
A SVM calculation prepared on a lot of preparing examples,each set apart to one or other of the two classes assembles a model that assigns out new examples(test) to one of the two classes,making it a binary,linear,non-probabilistic classifier. A SVM model speaks to

precedents as information focuses in space which are mapped to such an extent that the models having a place with various classes are isolated by an unmistakable edge which is as wide as possible.New models are anticipated to have a place with one of the two classes by mapping them into a similar space dependent on which side of edge they fall. Other than performing direct order, SVMs are likewise equipped for performing non-straight arrangement by certainly mapping their contributions to high-dimensional element spaces.A support-vector machine develops a hyperplane or set of hyperplanes in a high or unending dimensional space, which can be utilized for regression,classification, or different assignments like anomalies detection.Intuitively,the hyperplane that has the biggest separation to the closest preparing information purpose of any classification accomplishes great separation,since for the most part the bigger the gap,the lower is the speculation mistake of the classifier.

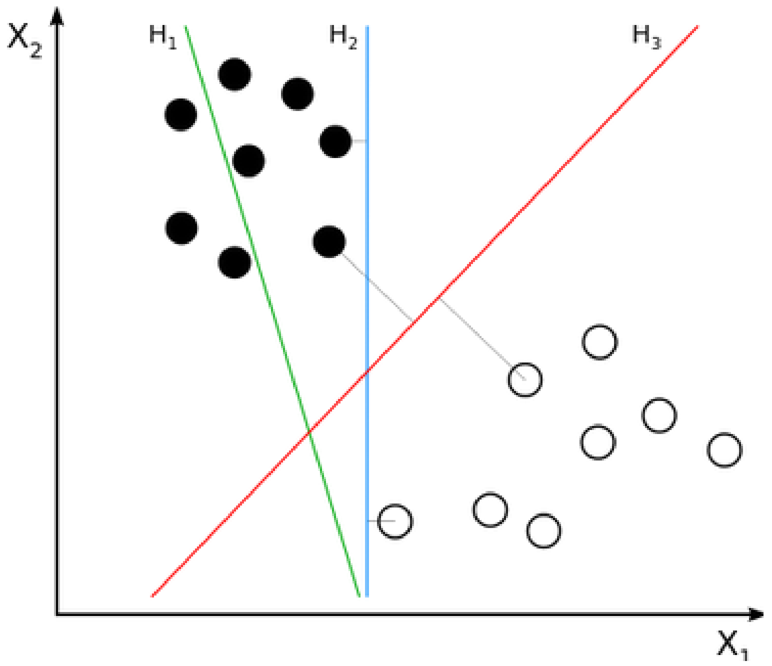


Figure 1: H1 doesn't differentiate the classes. H2 does,by a small margin. H3 differentiates them with the maximal gap[5]

3.3 Random Forest Classifier

Random Forest(also known as Random Decision Forest) are learning strategies which includes a gathering of decision trees to perform regression,classification and other tasks.Individual trees are prepared and irregular forests outputs that class which is the mode of the classes(in instance of order) or mean(in instance of relapse) of individual trees.Random Forest assists individual trees by restoring issue of overfitting in their case. Decision trees can perform different AI tasks and are in this manner popular.Learning of trees verges on filling in as an off-the-rack strategy for information mining fundamentally in the fact that it produces recognizable models,is strong to avoidance of irrelevant features,is invariant under scaling and different alterations of highlight values.But, they are infrequently accurate.
In particular, trees that are turned out to be profound will all in all adjust entirely unusual models: they overfit their readiness sets, for instance have low bias, yet amazingly high change. Random Forests are a technique for averaging different significant decision trees, arranged on different bits of a comparable planning set, with the goal of lessening the variance.This goes to the drawback of a little addition in the bias and some loss of interpretability, yet generally hugely helps with the performance of the model.

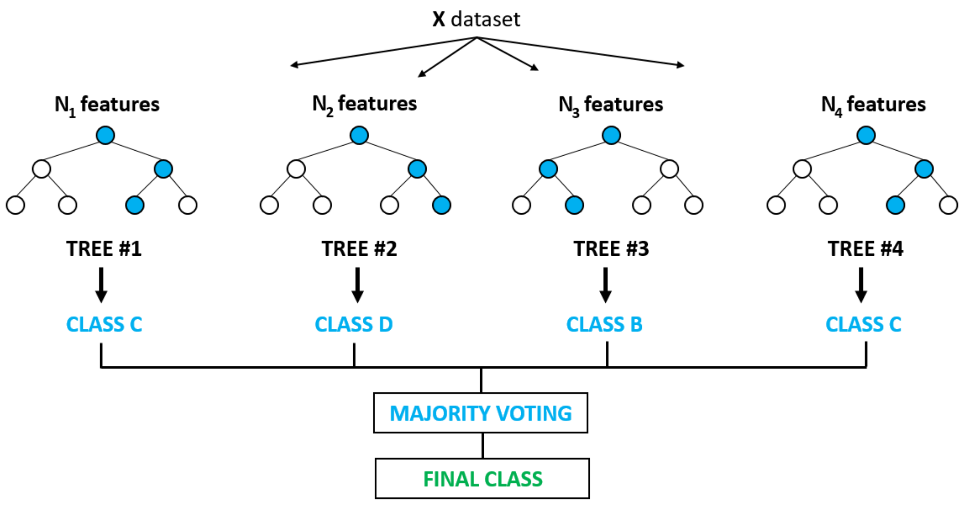


Figure 2: Working of Random Forest Classifier[6]

3.4 Multi Layer Perceptron(MLP)

A Multilayer Perceptron (MLP) is a sort of feedforward artificial neural system and is made out of various perceptrons.Input,hidden and output layers are incorporated in a MLP.Input layer sees the signal,output layer settles on a prediction or makes the decision about the inputs and hidden layers are the ones that play out the calculations associated with instance of MLP.There can be various hidden layers. Except for hubs of input layer,each node is a neuron that utilizes non-straight initiation function.MLP utilizes backpropagation strategy for preparing which is a managed learning technique.
Multilayer perceptrons are frequently requred to solve supervised learning issues, they train on a lot of input output matches and figure out how to demonstrate the relationship (or conditions) between those data sources and outputs. Preparing includes altering the parameters, or the loads and inclinations, to limit mistake. Backpropagation is utilized to make those weights and bias alterations in respect to the error, and the error itself can be estimated in an assortment of ways, including root mean squared error (RMSE).

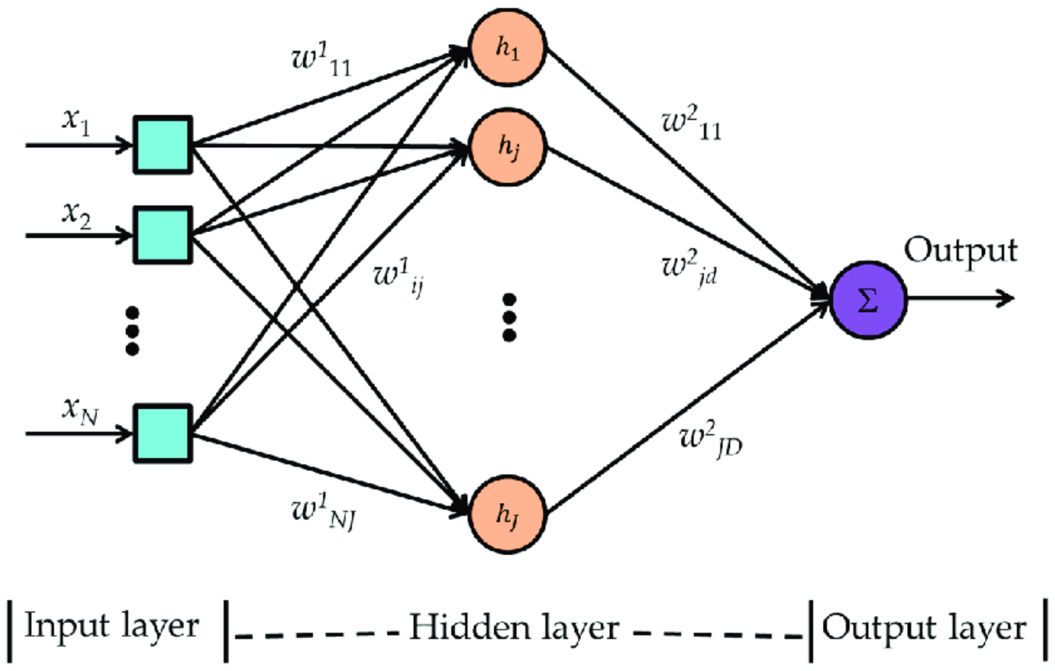


Figure 3: Working of MLP classifier[7]

3.5 Convolutional Neural Networks(CNN)

CNNs go under the classification of deep neural networks and are most generally and popularly connected for investigation of visual images. CNNs are multilayer perceptrons with regularization empowered. A convolutional neural framework contains an output and a input layer, similarly as various hidden layers. The concealed layers of a CNN involve convolutional layers, RELU,Tanh or Sigmoid layer for instance activation work, pooling layers to perform pooling activity, totally related layers and normalization layers. Preprocessing if there should be an occurrence of CNNs is significantly less than other image classification methods. From this we can induce that Convolutional Network learns channels that were hard-coded in customary algorithms.Achieving autonomy from before gained knowledge and human exertion in designing features is an incredible success.
While programming a convolutional layer in a neural system following traits must be remembered for each layer:

- Input is that of a tensor with shape (number of images) x (width of image) x (height of image) x (depth of image).
- Number of convolutional kernels.
  - Hyper-parameters are the width and height of kernel.
  - Depth of image must be equal to kernel depth.Convolution operation is applied by convolutional layer to the input and result is passed to the next layer.

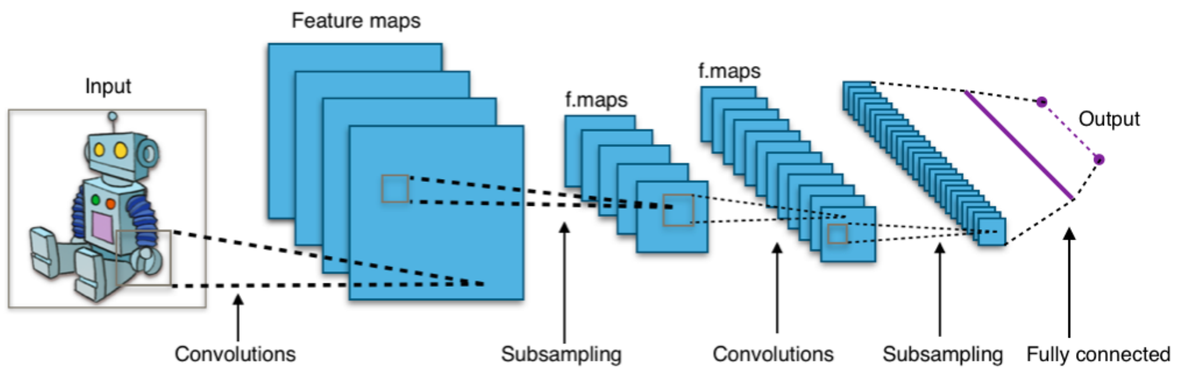


Figure 4: Working of CNN classifier[8]

3.6 Long Short Term Memory(LSTM)

LSTMs is a form of recurrent neural network (RNN) architecture utilized in the field of machine/deep learning. Dissimilar to standard feedforward neural systems, LSTM has input associations that make it a "universally useful computer".It possesses the ability to not just process single information points, (as an example, pictures), yet in addition whole groupings of information, (for example, discourse or video). A typical LSTM unit is made of a yield entryway, a cell, an information door and an overlook door. This cell recollects values over irregular timeframes and the three entryways direct the stream of data into and out of the cell.
LSTMs have favorable position over ordinary feed-forward neural systems and RNN from numerous points of view. This is expected due to their property of specifically recalling groupings for long time interims. LSTM networks are well-suited to classifying, processing and giving predictions based on time sequence data, since there can be lags of unknown duration between important events in a time series. LSTMs were made to deal with the evaporating and detonating slope issues that can be experienced while getting ready customary RNNs. A RNN utilizing LSTM units can be prepared in a directed manner, on a lot of training sequences, which utilizes an enhanced algo, similar to gradient descent, joined with backpropagation through the flow of time to figure the slopes required amid the advancement procedure, so as to change each weight of the LSTM organize in extent to the error derivative(at the output layer of the LSTM organize) as for relating weight.

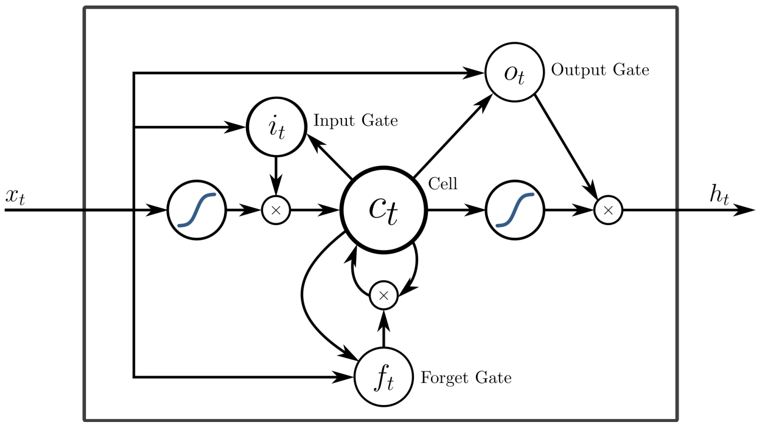


Figure 5: Working of LSTM[9]

Chapter 4

Experimental Setup and Results Analysis

We have followed a systematic approach to extract emotions from speech signals. Firstly we have procured an apt dataset and then processed it according to our requirements. Then a judicious amount of knowledge was required of the working of libraries in python to extract and pad the speech signals.Also the utility of various machine learning algorithms was stacked up against the results obtained.Efforts have been made for improving the classifier's accuracy by tweaking the algorithm according to our needs. The following are the datasets, libraries, frameworks, algorithms and techniques which we have utilized to establish the working system of our project:

4.1 Dataset

4.1.1 Berlin's Emotional Speech Database

We Have used Emo-DB dataset in our experiment. Dataset contains various emotions like happy,angry,anxious,fearful,bored,disgusted and neutral spoken by actors in about 500 utterances. Utterances can be chosen from 10 different actors and 10 different texts.
Naming of utterances is done according to following scheme:

- Speaker's number: Positions 1-2
- Code for text: Positions 3-5

- Emotion: Position 6
- a,b,c.. is the numbering given to more than one version of audio signal: Position 7

  Illustration : 15b09Fa.wav is the audio file from Speaker 15 speaking text b09 with the emotion "Freude".

Information about the speakers

- Speaker 03 is 31 years old and is male
- Speaker 08 is 34 years old and is female
- Speaker 09 is 21 years old and is female
- Speaker 10 is 32 years old and is male
- Speaker 11 is 26 years old and is male
- Speaker 12 is 30 years old and is male
- Speaker 13 is 32 years old and is female
- Speaker 14 is 35 years old and is female
- Speaker 15 is 25 years old and is male
- Speaker 16 is 31 years old and is female

4.2 Software And Libraries Used

4.2.1 Anaconda

Anaconda is open-source circulation of Python and R programming languages for logical computing (data science, AI applications, substantial scale information preparing, predictive analysis and so on) that intends to simplify package executives and deployment. Package variants are overseen by the package management framework conda.The Anaconda distribution is utilized by more than 13 million clients and incorporates in excess of 1400 mainstream data science packages appropriate for Windows, Linux, and MacOS.It gives substantial determination of bundles and business support. It is a domain chief, which gives the office to make diverse python conditions, each with their own settings.

4.2.2 Python 3.0

Python was developed by Guido van Rossum in early 1990s and its latest version is 3.7.1, we can simply call it as Python3. Python 3.0,released in 2008 is an interpreted language which means that the interpreter will check the code line by line and the code is not fully compiled. Python is broadly utilized for AI calculations on account of it contains unique libraries in particular scipy and numpy which extraordinary for straight variable based math and becoming more acquainted with bit strategies for AI. The language is incredible to utilize when working with AI calculations and has simple syntax relatively.

4.2.3 Scikit-learn

Scikit-learn gives a scope of managed and unsupervised learning calculations through a steady interface in Python. It is authorized under a permissive simplified BSD permit and is conveyed under numerous Linux distributions, empowering scholastic and business use. The library is based upon the SciPy (Scientific Python) that must be introduced before you can utilize scikit-learn. This stack incorporates:

- NumPy: Base n-dimensional array package
- SciPy: Scientific library for computing
- Matplotlib: Used for 2D/3D plotting comprehensively

- Pandas: Used for various data structures and analysis

Scikits are the extensions or modules for SciPy care.The module provides various learning algorithms and is named scikit-learn.

4.2.4 Keras

Keras,which runs upon Tensorflow is a Neural Network library which is Open Source. It has been designed such that it is fast,useful,easy to use and modular. Low level computations are not handled by Keras and it makes use of another library called Backend to perform such computations. Hence Keras acts as a high-level API wrapper for the low-level API's and is capable of running on top of CNTK,TensorFlow or Theano.

4.2.5 NumPy

NumPy is used for processing of arrays and is a general purpose package.It provides various tools for working with these arrays and a high-performing array-object of multiple dimensions. It is the fundamental package for scientific computing with Python. Various important features included in numpy are: Various broadcasting function tools of high complexity for integrating Fortran and C/C++ code, useful linear algebra, Fourier transform, a powerful n-dimensional array object and random number capabilities.

4.3 Models Used

- **Linear Support Vector Machine**

  Support Vector Machine chooses that hyperplane which maximizes the margin between the classes.For more than two classes they adopt one-vs-all classification.

  ```
  class SVM(MLModel):
      def _init_(self, **params):
          prams['name'] = 'SVM'
          super(SVM, self)._init_(**params)
          self.model = LinearSVC(multi_class='crammer_singer')
  ```

- **Random Forest Classifier**

  These classifers help to improve problem of overfitting which are faced by traditional decision trees.They are meta estimators which train a number of decision trees on parts of dataset and use their combined result to perform prediction.

  ```
  class RF(MLModel):
      def _init_(self, **params):
          params['name'] = 'Random Forest'
          super(RF, self)._init_(**params)
          self.model = RandomForestClassifier(n_estimators=30)
  ```

- **Multi Layer Perceptron**

  Here we have used the logistic sigmoid activation function.Number of hidden layer nodes have been chosen to be 512 and we have adopted a batch size of 32 samples.

  ```
  class NN(MLModel):
      def _init_(self, **params):
          params['name'] = 'Neural Network'
          super(NN, self)._init_(**params)
          self.model = MLPClassifier(activation='logistic',
          verbose=True,hidden_layer_sizes=(512,),batch_size=32)
  ```

- **Long Short Term Memory**

  In this model initially we have added a lstm layer of 128 units,then we have applied dropout regularization with a dropout value of 0.5.After that we have applied a dense layer having 32 units with relu activation function followed by a dense layer having 16 units with tanh activation function.Finally we have used a softmax layer for the output.

  ```
  self.model = Sequential() #declaring model to create sequence of layers
  self.model.add(lstm(128,input_shape=(self.input[0],self.input[1]) #adding lstm layer with 128 nodes
  self.model.add(Dropout(0.5)) #Applying "dropout regularization"
  self.model.add(Dense(32, function_activation='relu')) #Using "dense" layer with "relu" activation function
  self.model.add(Dense(16, function_activation='tanh')) #Using "dense" layer with "tanh" activation function
  self.model.add(Dense(num_classes, function_activation='softmax')) #Using "dense" layer with "softmax" activation function
  self.model.compile(loss='binary_crossentropy', optimizer='adam',
  metrics=['categorical_accuracy']) #Compiling model using Adam's optimization algorithm
  ```

- **Convolutional Neural Networks** In this model initially we have added a Conv2D layer with 8 filters and filter size of (13,13) having relu activation followed by a MaxPooling2D layer. Finally we have used a softmax layer for the output.

  ```
  self.model = Sequential()
  self.model.add(Conv2D(8, (13, 13),input_shape=(self.input[0],self.input[1], 1)))
  #Using "Convolutional Layer" with 8 filters and filter size of (13,13)
  self.model.add(BatchNormalization(axis=-1)) #Using "Batch Normalization"
  self.model.add(Activation('relu'))#Applying relu activation function
  self.model.add(MaxPooling2D(size_pool=(2, 1)))
  self.model.add(Dense(num_classes, activation='softmax'))#Using "MaxPooling layer"
  ```

```
with "kernel" size of (2,1)
self.model.compile(loss='binary_crossentropy', optimizer='adam',
metrics=['categorical_accuracy'])
```

4.4 Experiment

4.4.1 Preprocessing

The Emo-db database before being fed into our classifiers needs to be modified in order to obtain the desired output. Firstly, we need to divide the datasets into train and test and then start the proceedings. MFCC imposes a maximum length of the 32,000 samples for input, thus we have to pad the speech wave forms accordingly. If the length is less than the maximum length, the remainder is divided into two halves and padded at the start and end of the speech signal. If the length is more than maximum length, then the extra portion is removed from the start and end.

```
mslen = 32000#Empirically calculated for the given dataset
# pad the signals to have same size if lesser than required else slice them
if s_len <mslen:
    pad_len = mslen - s_len
    pad_rem = pad_len % 2
    pad_len //= 2
    signal = np.pad(signal, (pad_len, pad_len + pad_rem), 'constant', constant_values=0)
else:
    pad_len = s_len - mslen
    pad_len //= 2
    signal = signal[pad_len:pad_len + mslen]
```

4.4.2 Training

We have divided our dataset into training and test set as per 80:20 ratio.For deep learning models class labels has been converted to categorical data. For training of deep learning models we have devoted a validation set of 20% from training set and have made use of callbacks for early stopping to avoid overfitting. Training of deep learning models takes more time since there are many complex operations which need to be performed in their case. We have performed training in the following way:

```
def training(self,train_x,train_y,val_x=None, val_y=None):
    for i in range(50):
        permute = np.random.permutation(len(train_x))
        train_x = train_x[permute]
        train_y = train_y[permute]
        cnt = cnt+1
        early_stopping=EarlyStopping(monitor='val_categorical_accuracy',mode='max')
        self.model.fit(train_x,train_y,batch_size=32, num_of_epochs=4,validation_split=0.2,
        callbacks=[early_stopping])
        loss, accuracy = self.model.evaluate(val_x, val_y)
        print("accuracy",accuracy)
        if accuracy >accuracy_best:
            accuracy_best = accuracy
        self.trained = True
```

4.4.3 Testing

After training various classification models on our training data we need to test our model on test set. Here we plot confusion matrix for our models so that we can evaluate our model on various measures like Accuracy, Precision, Recall, F1-score.

```
def evaluate(self, test_x, test_y):
    pred_y = self.predict(test_x)
    print('Obtained_Accuracy',accuracy_score(y_pred=pred_y, y_true=test_y))
    cm = confusion_matrix(y_pred=pred_y, y_true=test_y)
    plot_confusion_matrix(cm,normalize=False,target_names =['Neutral', 'Angry', 'Happy',
     'Sad'],title = "Confusion Matrix")
```

4.4.4 Comparison of Results

Following conventions have been used here:

- Number of examples is denoted by n.
- The actual truth label of the i th example is denoted by Y_i.
- The ith example is denoted by x_i.
- The predicted labels for the ith example is denoted by $H(x_i)$.

$$Precision = (1/n) \sum_{i=1}^{n} \frac{|Y_i \bigcap H(x_i)|}{|H(x_i)|} \tag{1}$$

Precision of a particular class refers to what portion of the predicted part are correct. The numerator discovers what number of names in the anticipated vector has regular with the ground truth, and the proportion processes, what number of the anticipated genuine names are quite the ground truth.Intuitively, a high precision for a class implies that if our models foresee that class, it is in all respects prone to be valid. A high precision model will be valuable in those circumstances where we need a high trust in our expectation (for instance in medical diagnosis).

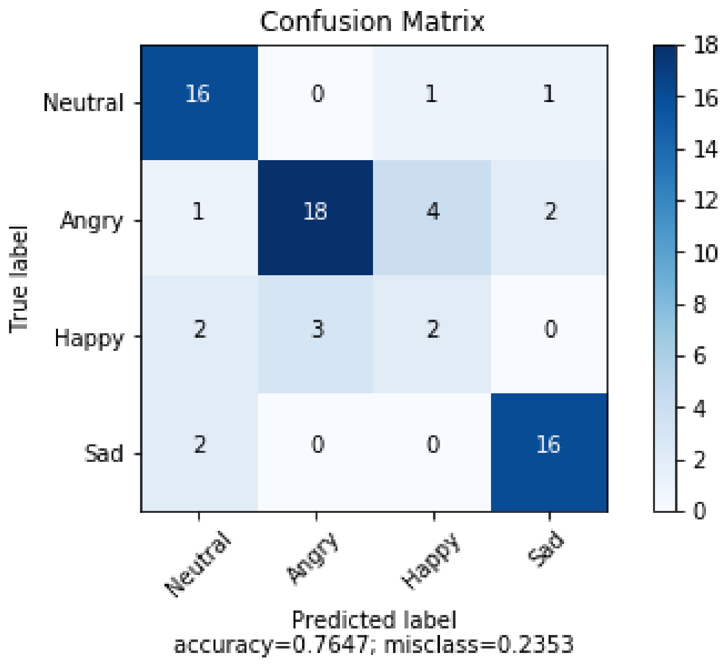
$$Recall = (1/n) \sum_{i=1}^{n} \frac{|Y_i \bigcap H(x_i)|}{|Y_i|} \tag{2}$$

Recall of a particular class refers to what portion of the actual labels were correctly predicted. The numerator discovers what number of names in the anticipated vector has normal with the ground truth (as above), at that point finds the proportion to the quantity of real marks, consequently getting what division of the real names were predicted.If recall is high, it implies that our models figures out how to recover most occurrences of that class. Getting high recall is extremely simple, it's adequate to state that everything matches that class, and you can make sure that every one of the components are recovered.

$$F1 - score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{3}$$

F1-score is the harmonic mean of precision and recall, and acts as a combined measure of the two.

- *Linear Support Vector Machine(SVM)*

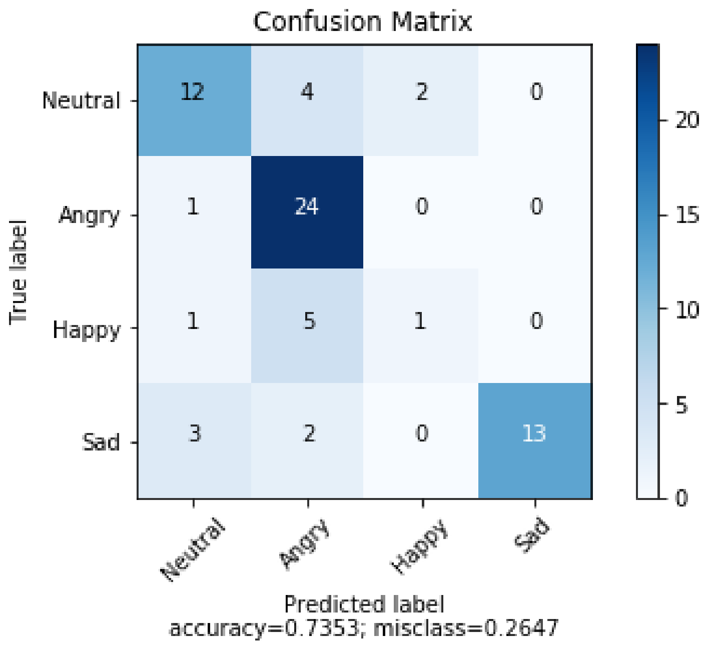


Class	**Precision-score**	**Recall-score**	**F1-score**
Neutral	0.762	0.888	0.820
Angry	0.857	0.720	0.944
Happy	0.286	0.286	0.286
Sad	0.842	0.888	0.864
Average	0.687	0.696	0.728

$$Accuracy = \frac{16 + 18 + 2 + 16}{68} = 76.47\%$$

Angry emotion state recorded the highest Precision and F1-score, because it's easier to get the differentiating hyperplane in this case as the tonal quality and pitch differs well when compared with other emotion states. Linear SVM has the advantage of non linear classification by mapping into higher dimensional feature space in order to match the shape of the features extracted from MFCC.

- *Random Forest Classifier*

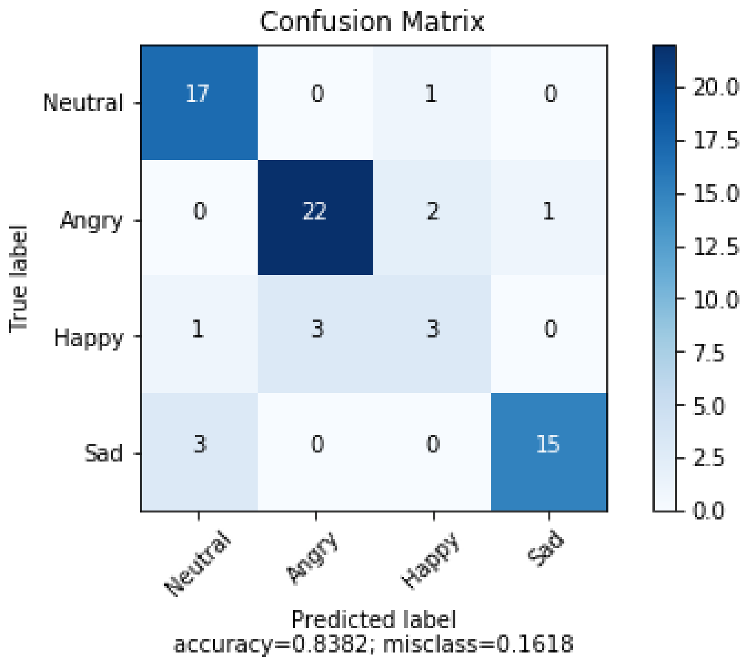


Class	Precision-score	Recall-score	F1-score
Neutral	0.706	0.667	0.686
Angry	0.686	0.960	0.800
Happy	0.333	0.143	0.200
Sad	1.000	0.722	0.838
Average	0.681	0.623	0.631

$$Accuracy = \frac{12 + 24 + 1 + 13}{68} = 73.53\%$$

Random Forest Classifier is invariant to scaling and the output is based on majority voting. This enabled sad emotion state to achieve 100% precision as sad emotion generally has a lower pitch and there were 13 predictions, all of which were correct. However, the happy one was on the lower side because it generally gets confused with neutral and sad tone, thus the recall value is pretty poor in random forest. A reasonable accuracy was obtained because the variance was reduced to prevent over-fitting and decision trees are a very good learner of complex features.

- *Multi Layer Perceptron(MLP)*

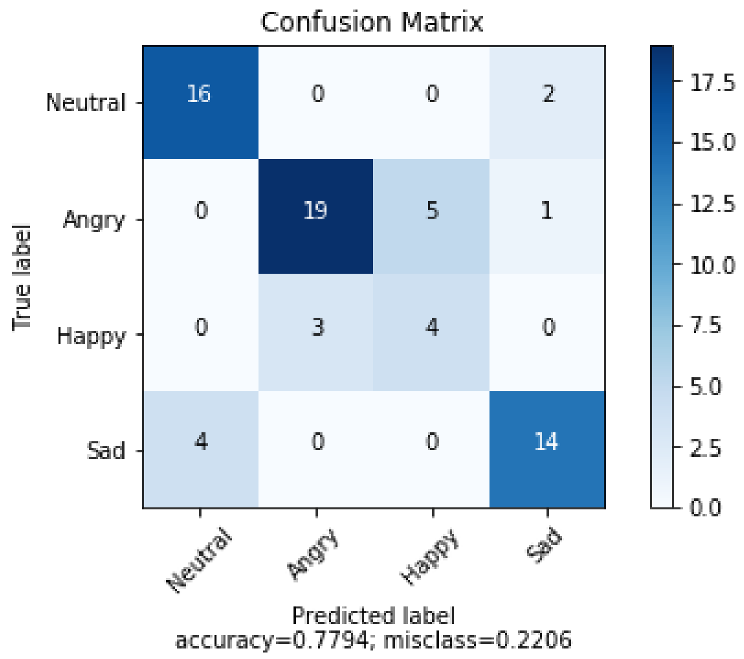


Class	**Precision-score**	**Recall-score**	**F1-score**
Neutral	0.809	0.944	0.871
Angry	0.880	0.880	0.880
Happy	0.500	0.428	0.461
Sad	0.938	0.833	0.882
Average	0.782	0.771	0.774

$$Accuracy = \frac{17 + 22 + 3 + 15}{68} = 83.82\%$$

Multi Layer Perceptron recorded second highest Accuracy and F1-score as it uses back-propogation and feedforward propogation. The amount of hidden layers and the weights greatly determine the performance of MLP and the fact that it uses deep learning techniques gives it a good edge on differentiating emotion states and identifying composite patterns well.

- *Long Short Term Memory(LSTM)*

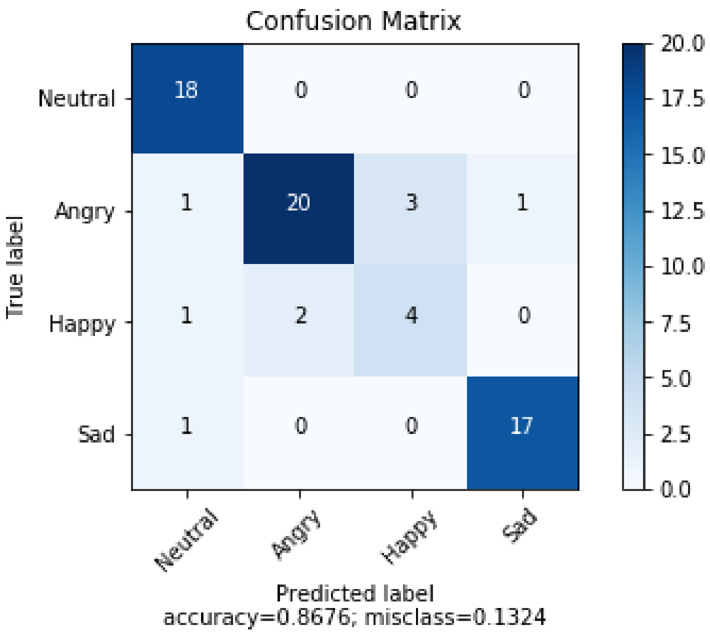


Class	Precision-score	Recall-score	F1-score
Neutral	0.800	0.888	0.842
Angry	0.864	0.760	0.809
Happy	0.444	0.571	0.500
Sad	0.823	0.777	0.799
Average	0.733	0.749	0.738

$$Accuracy = \frac{16+19+4+14}{68} = 77.94\%$$

LSTMs selectively remember patterns for long durations of time and therefore they have an edge over conventional feed-forward neural networks.LSTMs are essentially a time-series model which are nonlinear and the non-linearity is learnt from the data. We have obtained good precision values for Neutral, Angry and sad emotions as LSTM is great tool for anything that has a sequence, and the speech signals tend to follow different patterns for different emotional states.

- *Covolutional Neural Networks(CNN)*



Class	Precision-score	Recall-score	F1-score
Neutral	0.857	1.000	0.923
Angry	0.909	0.800	0.851
Happy	0.571	0.571	0.571
Sad	0.944	0.944	0.944
Average	0.820	0.829	0.823

$$Accuracy = \frac{18 + 20 + 4 + 17}{68} = 86.76\%$$

Using Convolutional Neural Networks(CNNs) we obtained the highest Accuracy and F1-score proving that it performed best among our classification algorithms.CNNs are best suited for multimedia applications.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

In this project we have successfully applied emotion recognition to speech signals obtained from Emo-db dataset and have achieved an accuracy of 86.76% while using Convolutional Neural Networks(CNN) as our classifier. Four emotion states(Anger, Happy, Sad and Neutral) have been analyzed and tried to be predicted in the speech signals using the features extracted by MFCC. We have noticed that Happy emotion has recorded the least in precision and recall values across all classifiers, indicating that people express happiness in numerous ways which is more individualistic and cannot be ascertained a pattern. However, Angry, Neutral and Sad have consistently been predicted well showing that these emotion states have followed vocal and acoustic pitch patterns that can be identified well. CNN proves to be marginally better as in deep learning method it includes layer processing with less time and the weights are assigned in a way to perform with better accuracy. We also have seen why MFCC is widely used in automatic speech and speaker recognition, it's capability to accurately represent and extract the useful features have been demonstrated. Thus, emotion recognition from speech signals still has a long way to go, but these results can't be undermined.

5.2 Future Work

We can extend the emotion states to fear and disgust as well, thereby providing more insight into how our vocal undertones affect and impose the emotional states. After developing an advanced version of emotion detection in speech, we can deploy it into real-time models where they can be utilized in various forms. As an example, in suicide helplines or while monitoring police emergency phone lines we can find the emotion of the person on call using our project and put up a trigger line to start a course of actions which may end up saving people's life and assets.It can also be applied to instance of manual driving where the driver is taking some VIP person with him and so his emotions need to be monitored.

References

[1] E. Ramdinmawii, A. Mohanta and V. K. Mittal, "Emotion recognition from speech signal," TENCON 2017 - 2017 IEEE Region 10 Conference, Penang, 2017, pp. 1562-1567.doi: 10.1109/TENCON.2017.8228105

[2] M. Ghai, S. Lal, S. Duggal and S. Manik, "Emotion recognition on speech signals using machine learning," 2017 International Conference on Big Data Analytics and Computational Intelligence (ICBDAC), Chirala, 2017, pp. 34-39. doi: 10.1109/ICBDACI.2017.8070805

[3] Pal Singh, Parwinder. (2014). "An Approach to Extract Feature using MFCC", IOSR Journal of Engineering.

[4] S. Basu, J. Chakraborty, A. Bag and M. Aftabuddin, "A review on emotion recognition using speech," 2017 International Conference on Inventive Communication and Computational Technologies (ICICCT), Coimbatore, 2017.

[5] Duan, Kai-Bo, Keerthi, S. Sathiya (2005). "Which Is the Best Multiclass SVM Method? An Empirical Study". Multiple Classifier Systems. LNCS. 3541.

[6] Shi, T., Horvath, S. (2006). "Unsupervised Learning with Random Forest(RF) Predictors". Journal of Computational and Graphical Statistics.

[7] R. Collobert and S. Bengio (2004)." Links between Perceptrons, MLPs and SVMs". Proc. Int'l Conf. on Machine Learning (ICML).

[8] Durjoy Sen Maitra, Ujjwal Bhattacharya and S.K. Parui, "CNN based common approach to handwritten character recognition of multiple scripts in Document Analysis and Recognition (ICDAR)", 2015 13th International Conference.

[9] Klaus Greff,Rupesh Kumar Srivastava, Jan Koutnk, Bas R. Steunebrink and Jrgen Schmidhuber (2015). "LSTM: A Search Space Odyssey". IEEE Transactions on Neural Networks and Learning Systems.

[10] S. Lugovi, I. Duner and M. Horvat. "Techniques and Applications of Emotion Recognition in Speech". Zagreb University of Applied Sciences, Department of Computer Science and Information Technology.