Classification of Emotion Detection using Deep Neural Network

Shivam Malik[1], Brijmohan Singh[2], Himanshu Chauhan[3]

[1-3]Department of CSE, College of Engineering Roorkee, Roorkee – 247667, Uttarakhand, India

Abstract: Emotions are subjective, people would interpret it differently. It is hard to define the notion of emotions. Annotating an audio recording is challenging. We label a complete sentence. There are lots of audio data can be achieved from films or news therefore collection of data is complex. However, both of them are biased since news reporting has to be neutral and actors' emotions are imitated. It is hard to look for neutral audio recording without any bias. Labelling data require high human and time cost. Unlike drawing a bounding box on an image, it requires trained personnel to listen to the whole audio recording, analysis it and give an annotation. The annotation result has to be evaluated by multiple individuals due to its subjectivity. In particular, we are presenting a classification model of emotions elicited by speeches based on deep neural networks (CNNs). For the purpose, we focused on the audio recordings available in the Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) dataset. Additional training and testing we are done on the TESS dataset [13] which contains a set of 200 sentences spoken by two actresses in 7 emotions. The model has been trained to classify eight different emotions (neutral, calm, happy, sad, angry, fearful, disgust, surprise) which correspond to the ones proposed by Ekman plus the neutral and calm ones. We considered as evaluation metric the F1 score, obtaining a weighted average of 0.91 on the test set and the best performances on the "Angry" class with a score of 0.95. Our Deep Neural Network model are trained on the dataset using feature extraction for each sample. The average precision obtained is 0.92 and recall is 0.91. The f1-score of the model is 0.91 with average support of 1633 and the accuracy of random forest model is 0.91.

***Keywords*: emotion detection, natural language understanding, sentiment analysis, deep learning, machine learning, classification, mel-frequency cepstral coefficients, cnn, ravdess.**

1. INTRODUCTION

Over the years a lot of research has gone into understanding speech emotion recognition from a multi-disciplinary standpoint. Researchers from neuroscience area understand the way brain perceives the input stimuli by processing the raw data and applying the knowledge stored in the brain to reflex the various actions. Computational intelligence researchers provide tools to justify the knowledge gained from the neuroscientists in a mathematical solution while linguists would rather view speech that can provide emotional knowledge through semantic and syntactic analysis of the speech. These combinations of interdisciplinary researches have shown tremendous potential in understanding speech emotion. Scherer (2003) provides an overview of the various design paradigms on the subject using a modified Brunswick's functional lens model of perception. There have been many other attempts as well at using the spectral analysis and Hidden Markov Model for identifying and recognizing emotions as described in Ververidis and Kotro-poulos (2006), Womack and Hansen (1999) and Zhou et al. (2001).

Humans interact with their environment using many different sensing mechanisms. Detecting an unpleasant state during the task and intervening the process is possible with real time a detective systems. In human computer interaction, the main task is to keep users level of satisfaction as high as possible. A computer with a detective properties could detect the user's emotion and could develop a counter response to increase user satisfaction. Speech and gesture recognition are the most popular effective computing topics. Speech and gesture recognition are possible with passive sensors. While in the new age of information security there is an unease in granting full video sensor access but audio based devices are already well entrenched in the market and hence an excellent avenue for exploring emotion detection.

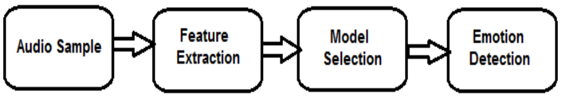


Figure 1: Data Flow Model

In a typical model it is important to decide the features for extraction. These are mostly based on a series of robust mathematical models. The same exists for speech analysis. Vocal tract information like formant frequency, bandwidth of formant frequency and other values may be linked to a sample. There is a wide variety of options for parametrically representing the speech signal in a machine understandable way so that statistical analyses can be performed on it to arrive at informed results. Some of these techniques are: Linear Prediction Coding (LPC); Mel-Frequency Cepstral Coefficients (MFCC); Linear Predictive Cepstral Coefficients (LPCC); Perceptual Linear Prediction (PLP); and Neural Predictive Coding (NPC). Mel Frequency Cepstral Coefficients (MFCC) is a popular technique because it is based on the known variation of the human ear's critical frequency bandwidth. MFCC coefficients are obtained by de-correlating the output log energies of a filter bank which consists of triangular filters, linearly spaced on the Mel frequency scale. Conventionally an implementation of discrete cosine transform (DCT) known as distributed DCT (DCT - II) is used to de-correlate the speech as it is the best available approximation of the Karhunen-Loeve Transform (KLT). (Sahidullah and Saha; 2009).MFCC data sets represent a melodic cepstral acoustic vector (Barbu; 2009), (Wang et al.; 2006). The acoustic vectors can be used as feature vectors. It is possible to obtain more detailed speech features by using a derivation on the MFCC acoustic vectors. You can obtain higher order MFCC coefficients as well as log energy scale values for them which are all valid features to aid in classification. The next step after the feature selection is the model selection. There have been a vast array of models that have been explored for emotion detection. We will focus on 3 main models namely: Decision Tree, Random Forest and DNN.

The rest of this paper is arranged in the following order. The Related work section deals with relevant papers from the community which inspire us in our current paper. The Methodology and design section will explain in detail the experimental setup for the purpose of reproducibility. The implementation will detail the exact steps undertaken to clean and create the data as well as the models that are going to be used. The evaluation and future work section will expand on the results obtained from the models and how to expand them for better performance in the future.

2. REVIEW WORK

Any consumer specific device can tailor make the interaction with its user if it can get access to their current emotional make-up. This will help create a more seamless and enriching interactive experience. This is why we see a host of new research in this area.

Speech based emotion recognition has a few core areas that are important:

- Feature extraction.
- Classification algorithm.

We will focus the related work in each of these areas and showcase the advantages and disadvantages.

2.1 Feature extraction

In their paper Gupta et al. (2014) enumerate that in recent times various speech feature extraction methods have been proposed. Diverse methods are differentiated by the ability to use most information about human speech processing perception by considering distortions and by the length of the observation window. The speech is highly redundant due to human physiology and has a variety of speaker-dependent features such as pitch, speaking rate, frequency and accent. In the paper they have employed a pitch energy and MFCC based feature selection.

Zeng et al. (2008) in their work, also showcase the MFCC based features. They use it because of its low complexity, better ability to extract the feature from speech, efficient technique and also has the advantage like anti-noising etc.

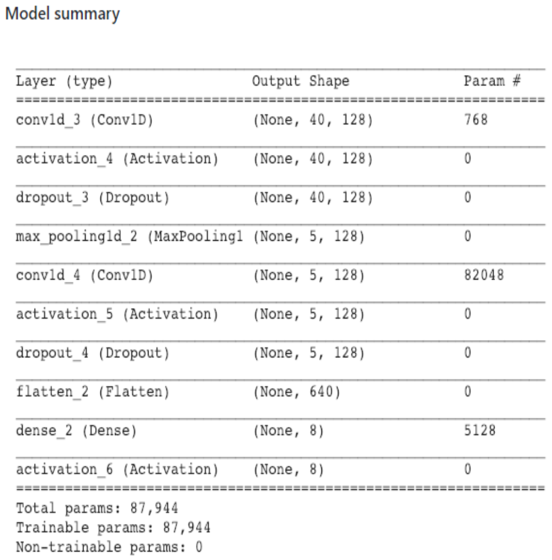
After suppressing vocal tract (VT) characteristics, excitation source signal is obtained from speech. This is achieved by first predicting the vocal tract information using linear prediction coefficients from speech signal and then separating it by inverse filter formulation. The resulting signal contains mostly the information about the excitation source and is known as linear prediction residual (Makhoul; 1975). The paper by Shashidhar et al. (2012) explores the concept of using LPC for detection of emotions. There are few other papers that use this possibly because it is seen as an error signal. The concept of using pitch and energy has been explored in the work written by Schuller et al. (2003).An analysis is done on the contours of pitch and energy since they have a well-known capability to carry a large amount of information considering a user's emotion and. In order to calculate the contours, a Hamming window function is used every 10 seconds to analyse the frames of the speech signal. Energy value is calculated by the logarithmic mean energy within a frame. By using average magnitude difference function (AMDF), the pitch contour is achieved.

2.2 Classification Algorithm

The model of classification of emotions here proposed is based on a deep learning strategy based on convolutional neural networks (CNN) and dense layers [8]. The key idea is considering the Mel-frequency cepstral coefficients [1], [10] (MFCC), commonly referred to as the "spectrum of a spectrum", as the only feature to train the model. MFCC is a different interpretation of the Mel-frequency cepstrum (MFC), and it has been demonstrated to be the state of the art of sound formalization in automatic speech recognition task [11]. The MFC coefficients have mainly been used has the consequence of their capability to represent the amplitude spectrum of the sound wave in a compact vectorial form. As described in [10], the audio file is divided into frames, usually using a fixed window size, in order to obtain statistically stationary waves. On the small frames obtained, the Discrete Fourier Transformation is applied, and only the logarithm of the amplitude spectrum is kept. The amplitude spectrum is normalized with a reduction of the "Mel" frequency scale. This operation is performed for empathizing the frequency more meaningful for a significant reconstruction of the wave as the human auditory system can perceive. For each audio file, 40 features have been extracted. The feature has been generated converting each audio file to a floating point time series. Then, a MFCC sequence has been created from the time series. The MFCC array has been transposed and the arithmetic mean has been calculated on its horizontal axis. The MFCC calculations are deeply explained in the article of Davis S. et al.[1] and in the book of Huang et al.[5]. In this field of research, many classification strategies have been presented in the last years. One of the system, proposed by Iqbal et al.[6] used Gradient Boosting, KNN and SVM to work on a granular classification on the RAVDESS dataset used in this work to identify differences based on gender with approximately between 40% and 80% overall accuracy, depending on the specific task. In particular, the proposed classifiers performed differently in different datasets but we considered only the work done on RAVDESS for the scope of this paper. In Iqbal et al.[6] work, three types of datasets have been created including only male recordings, only female recordings and a combined one. In RAVDESS (male) SVM and KNN have 100% accuracy in both anger and neutral, but in happiness and sadness Gradient Boosting performs better than SVM and KNN. In RAVDESS (female) SVM achieves 100% accuracy in anger as same as male part. SVM has overall good performance except in sadness. Performance of KNN is also good in anger and neutral like 87% and 100% respectively. In anger and neutral, Gradient Boosting performs poorly. KNN performance is very poor in happiness and sadness comparing with other classifiers. In male and female combined dataset, performances of SVM and KNN are really good in anger and neutral rather than Gradient Boosting. KNN's performance is really poor in happiness and sadness. Average performances of classifiers in male dataset are better than female dataset except SVM. In combined database, SVM get high accuracy than gender based datasets. Another approach presented by Jannat et al.[7] achieved 66.41% accuracy on audio data and more than 90% accuracy mixing audio and video data. In particular, given preprocessed image data that includes faces and audio waveforms, Jannat et al.[7] trained 3 separately deep networks: one network only on image data, another only on the plotted audio waveforms, and the third on both image and waveform data. One of the first approaches that used the RAVDESS dataset, but classifying only some of the emotions available, was published by Zhang et al.[16], reaching an overall accuracy score higher than the model proposed in this work but using less classes. Giving more details, in Zhang et al.[16] three shared emotion recognition models for speech and song have been proposed: a simple model, a single-task hierarchical model and a multi-task hierarchical model. The simple model creates a single classifier, independent of domain. The two hierarchical models use domain during training. The single task model trains a separate emotion classifier for each domain. The multi-task model trains a multi-task classifier to jointly predict emotion across both domains. In the testing phase, the testing data are separated based on the predicted domain. The data are analyzed using the classifier corresponding to the estimated domain. The work have been conducted adopting the directed acyclic graph SVM (DAGSVM) [14].

3. RESEARCH METHODOLOGY

The deep neural network designed for the classification task is reported operationally in Fig. 1. The network is able to work on vectors of 40 features for each audio file provided as input. The 40 values represent the compact numerical form of the audio frame of 2s length. Consequently, we provide as input a of size < number of training files > x40x1 on which we performed one round of a 1D CNN with a ReLu activation function [12], dropout of 20% and a max-pooling



Model summary

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv1d_3 (Conv1D)            (None, 40, 128)           768
_________________________________________________________________
activation_4 (Activation)    (None, 40, 128)           0
_________________________________________________________________
dropout_3 (Dropout)          (None, 40, 128)           0
_________________________________________________________________
max_pooling1d_2 (MaxPooling1 (None, 5, 128)            0
_________________________________________________________________
conv1d_4 (Conv1D)            (None, 5, 128)            82048
_________________________________________________________________
activation_5 (Activation)    (None, 5, 128)            0
_________________________________________________________________
dropout_4 (Dropout)          (None, 5, 128)            0
_________________________________________________________________
flatten_2 (Flatten)          (None, 640)               0
_________________________________________________________________
dense_2 (Dense)              (None, 8)                 5128
_________________________________________________________________
activation_6 (Activation)    (None, 8)                 0
=================================================================
Total params: 87,944
Trainable params: 87,944
Non-trainable params: 0
```

Figure 2. Detailed description of the architecture of the proposed classifier function 2 x 2.

The rectified linear unit (ReLu) can be formalized as g(z) = max{0,z}, and it allows us to obtain a large value in case of activation by applying this function as a good choice to represent hidden units.

Pooling can, in this case, help the model to focus only on principal characteristics of every portion of data, making them invariant by their position. We have run the process described once more by changing the kernel size. Following, we have applied another dropout and then flatten the output to make it compatible with the next layers. Finally, we applied one Dense layer (fully connected layer) with a softmax activation function, varying the output size from 640 elements to 8 and estimating the probability distribution of each of the classes properly encoded (0=Neutral; 1= Clam; 2= Happy; Sad=3; Angry=4; Fearful= 5; Disgust=6; Surprised=7).

3.1 Dataset

For the majority of our testing and training, we were using the RAVDESS dataset [12] which consists of 8 classes of human emotion. This dataset contains labeled files in 3 modality (fullAV, video-only and audio-only) and 2 vocal channels (speech and song) from male and female actors. Since our focus was on speech sentiment analysis, and also due to file size constraints, our models were trained on audio-only speech samples which consist of

Labels: total=1440, neutral=96, calm=192, happy=192, sad=192, angry=192, fearful=192, disgust=192, surprised=192.

These labels correspond in order to the confusion matrix in 3b. Additional training and testing were done on the TESS dataset [13] which contains a set of 200 sentences spoken by two actresses in 7 emotions. The dataset contains

Labels: total=2800, neutral=400, happy=400, sad=400, angry=400, fearful=400, disgust=400, surprised=400.

Compared to the previous dataset, the TESS dataset is cleaner in loudness and length which brings more consistency to training and testing. However, all the sentences in this dataset are led by the phrase "Say the word", which can be less universal and lack variety, which may eventually cause overfitting and thus harm the final performance. Hence TESS here was only applied as an additional dataset.

All the data were pre-processed and standardized sample by sample, as there were not too many samples, and standardizing across samples would make the data have little variance and hard to distinguish across classes. The normalization was based on loudness and length, for which we did zero padding to the heads and tails of all samples to make them length invariant.

3.2 Feature Extraction

Feature extraction is the next crucial step in our process. It is imperative to gather the right features as we cannot work directly with the audio le. This means the set of features we will select will be the representative of the original data in our models and this greatly enhances the role of the features in our research. Keeping this factor in mind we aim to extract almost all aspects which can give us information regarding the vocal characteristics of the speakers and samples. Below are the set of features we will be using and table shows and example row of the features.

MFCC: MFCC's are derived from the cepstral representation of the audio clip. They are derived by taking the fourier transform of the signal and mapping it to the mel scale. The take the discrete cosine transform of the powers of the mel log powers to obtain the MFCC signal (Ramirez et al.; 2018).

Chromagram from the waveform: Chroma features are an interesting and powerful representation for music audio in which the entire spectrum is projected onto 12 bins representing the 12 distinct semitones (or chroma) of the musical octave. Since, in music, notes exactly one octave apart are perceived as particularly similar, knowing the distribution of chroma even without the absolute frequency (i.e. the original octave) can give useful musical information about the audio { and may even reveal perceived musical similarity that is not apparent in the original spectra.

Mel scale Spectrogram: Computes a spectrogram on the basis of the Mel-scale. Spectral contrast of waveform: Each frame of a spectrogram S is divided into sub-bands. For each sub-band, the energy contrast is estimated by comparing the mean energy in the top quantile (peak energy) to that of the bottom quantile (valley energy). High contrast values generally correspond to clear, narrow-band signals, while low contrast values correspond to broad-band noise (Jiang, Lu, Zhang, Tao and Cai; 2002).

Tonal Centroid features: Computes the tonal centroid features as explained in Harte et al. (2006). This helps in understanding the harmonic change in audio signal.

3.3 The Proposed Model

The model of classification of emotions here proposed is based on a deep learning strategy based on convolutional neural networks (CNN) and dense layers [8]. The key idea is considering the Mel-frequency cepstral coefficients [1], [10] (MFCC), commonly referred to as the "spectrum of a spectrum", as the only feature to train the model. MFCC is a different interpretation of the Mel-frequency cepstrum (MFC), and it has been demonstrated to be the state of the art of sound formalization in automatic speech recognition task [11]. The MFC coefficients have mainly been used has the consequence of their capability to represent the amplitude spectrum of the sound wave in a compact vectorial form. As described in [10], the audio file is divided into frames, usually using a fixed window size, in order to obtain statistically stationary waves. On the small frames obtained, the Discrete Fourier Transformation is applied, and only the logarithm of the amplitude spectrum is kept. The amplitude spectrum is normalized with a reduction of the "Mel" frequency scale. This operation is performed for empathizing the frequency more meaningful for a significant reconstruction of the wave as the human auditory system can perceive. For each audio file, 40 features have been extracted. The feature has been generated converting each audio file to a floating point time series. Then, a MFCC sequence has been created from the time series. The MFCC array has been transposed and the arithmetic mean has been calculated on its horizontal axis. The MFCC calculations are deeply explained in the article of Davis S. et al.[1] and in the book of Huang et al.[5].

The deep neural network designed for the classification task is reported operationally in Fig. 1. The network is able to work on vectors of 40 features for each audio file provided as input. The 40 values represent the compact numerical form of the audio frame of 2s length. Consequently, we provide as input a of size < number of training files > x40x1 on which we performed one round of a 1D CNN with a ReLu activation function [12], dropout of 20% and a max-pooling function 2 x 2. The rectified linear unit (ReLu) can be formalized as g(z) = max{0,z}, and it allows us to obtain a large value in case of activation by applying this function as a good choice to represent hidden units. Pooling can, in this case, help the model to focus only on principal characteristics of every portion of data, making them invariant by their position. We have run the process described once more by changing the kernel size. Following, we have applied another dropout and then flatten the output to make it compatible with the next layers. Finally, we applied one Dense layer (fully connected layer) with a softmax activation function, varying the output size from 640 elements to 8 and estimating the probability distribution of each of the classes properly encoded (0=Neutral; 1= Clam; 2= Happy; Sad=3; Angry=4; Fearful= 5; Disgust=6; Surprised=7).

3.4 Classification Algorithms

There are different classification algorithms are exists. But we are using Deep Neural Network to implement problem statement. They have all been implemented in python using specialized libraries that offers parameter optimization and evaluation metrics.

3.4.1 Deep Neural Network

The DNN is an extension of the Artificial Neural Network (ANN) with more than one hidden layer between the input and output layer. The DNN is a very resource intensive and black box process. We cannot understand the underlying workings within the model. Only the results can be interpreted using the various metrics like accuracy and F-scores. The model has been put to use with great success in the research done by Han et al. (2014).

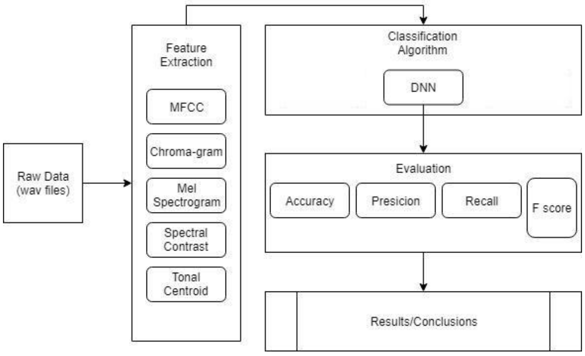


Figure 4. Process Architecture of Proposed Model

The DNN is implemented using the Keras package using Tensor flow as the back-end engine. A DNN is defined as a neural network that has more than one hidden layer between the input layers and output layers. We first have to one hot encode our label. One hot encoding converts the labels into a binary representation which will be used for classification. We have 4 Hidden layers in our model. The first layer has an input shape equal to the input data. The second layer has roughly twice the input shape of the first layer. The third layer has half of the input shape of the second layer and the fourth layer has an input shape half of the third layer. We use the Rectified Linear Unit (RelU) function as our activation function. Our optimization function is adamax. We train the model for 1000 epochs.

4. EVALUATION

Evaluation will be performed using various metrics like classification accuracy, precision, recall, F1-score. The F1 score reaches its best value as it tends to 1. We will compare and contrast the scores across our experiments to present conclusive evidence on which method gives the best results for detection of speech from audio samples.

The evaluation of the experiments are carried out using the following metrics:

True Positive (TP): A true positive is an outcome where the model correctly predicts the positive class.

True Negative (TN): A true negative is an outcome where the model correctly predicts the negative class.

False Positive (FP): A false positive is an outcome where the model incorrectly predicts the positive class.

False Negative (FN): A false negative is an outcome where the model incorrectly predicts the negative class.

Precision: The precision for a class is the number of true positives (i.e. the number of items correctly labeled as belonging to the positive class) divided by the total number of elements labeled as belonging to the positive class (i.e. the sum of true positives and false positives, which are items incorrectly labeled as belonging to the class).

$$\text{Precision} = \frac{TP}{(TP + FP)}$$

High precision means that an algorithm returned substantially more relevant results than irrelevant ones.

Recall: Recall is defined as the number of true positives divided by the total number of elements that actually belong to the positive class (i.e. the sum of true positives and false negatives, which are items which were not labeled as belonging to the positive class but should have been).

$$\text{Recall} = \frac{TP}{(TP + FN)}$$

High recall means that an algorithm returned most of the relevant results.

F-score: The F-beta score can be interpreted as a weighted harmonic mean of the precision and recall, where an F-beta score reaches its best value at 1 and worst score at 0.

$$F1 = \frac{2 \text{ x } (\text{Precision x Recall})}{(\text{Precision} + \text{Recall})}$$

Support: The support is the number of occurrences of each class.

Accuracy: Model accuracy in terms of classification models can be defined as the ratio of correctly classified samples to the total number of samples:

$$\text{Accuracy} = \frac{(TP + TN)}{(TP + TN + FP + FN)}$$

4.1 Evaluation of the Model

The evaluation of the proposed model has been carried out in order to investigate whether the model produces results of accuracy that are good enough to produce interesting considerations to be used in future work on the subject that include speeches in real noisy domains. Different models of classification have been evaluated beyond that proposed by us so that it is possible to generate baselines for the results obtained. As a first approach, a decision tree (DT) and a random forest (RF) classifiers with 1000 trees have been performed. The two models have been implemented using Scikit-learn 1 [13] Python library with default parameters.

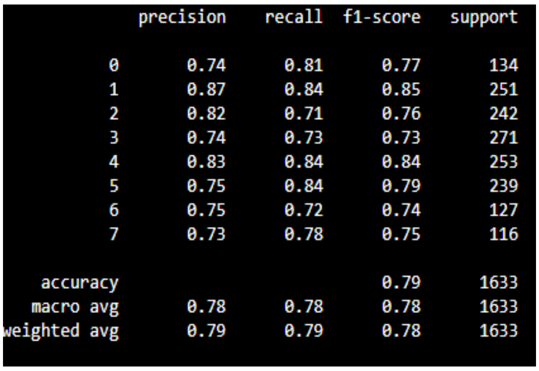
5. RESULTS AND DISCUSSION

The results obtained from the evaluation phase show the effectiveness of the model compared to the baselines and the state of the art on the RAVDESS and TESS dataset. In particular, Tab. I shows the values of precision, recall and F1 obtained for each of the emotional classes. These results show us that precision and recall are very balanced, allowing us to obtain F1 values distributed around the value 0.90 for almost all classes. The small variability of F1 results point out the robustness of the model that effectively manages to correctly classify emotions in eight different classes. The classes ”Sad” and ”Surprised” are the ones in which the model is less accurate, but this result does not surprise us because it is known in the literature that they are the most difficult classes to identify not only by speech but also while observing facial expression or analysing written text [15]. In order to evaluate the effectiveness of the classification of emotions proposed in this work, we decided to compare it with the results obtained from two baselines decision tree (DT) and random forest (RF) and the works of Iqbal et al.[6] and Zhang et al.[16]. The results shown in Tab II allow us to observe how the F1 values of our model are better than baselines and competitors on all classes except “Angry” and “Happy”. However, it is necessary to point out that the drop in performance is minimal, and the model works on four classes more than the model proposed by Iqbal et al.[6] and one class more than the model of Zhang et al.[16]. It is therefore well known that as the number of classes increases, the classification task becomes more complex and loses its accuracy. Nevertheless, the CNN-MFCC model proposed here manages to obtain a score of F1 that, on average, is equivalent to that of the two jobs we have been confronted with. A further index of model reliability can be found in Fig. 2 and Fig. 3. In the first one, it is possible to observe how the value of loss (error in the accuracy of the model) tends to decrease both on the test set and on the training set up to the 1000th epoch. The decrease is less evident from the 400th epoch but still perceptible. In Fig. 3, it is reported the average value of accuracy on all the classes that, to the contrary of the loss, increases with the increases of the ages. Such values of loss and accuracy do not differ much among the training and test dataset, allowing us to affirm that the model does not turn out to be overfitted while training. The consequence of this is, in fact, in line with the F1 scores previously observed.

Table I. Results of the model on the test set per each class

Emotion	precision	Recall	F1-score	support
Angry	0.93	0.91	0.92	134
Happy	0.92	0.93	0.92	251
Neutral	0.91	0.89	0.90	242
Sad	0.84	0.90	0.87	271
Calm	0.96	0.94	0.95	253
Fearful	0.92	0.91	0.91	239
Disgusted	0.95	0.93	0.94	127
Surprised	0.90	0.85	0.88	116
accuracy			0.91	1633
macro avg	0.92	0.91	0.91	1633
weighted avg	0.91	0.91	0.91	1633

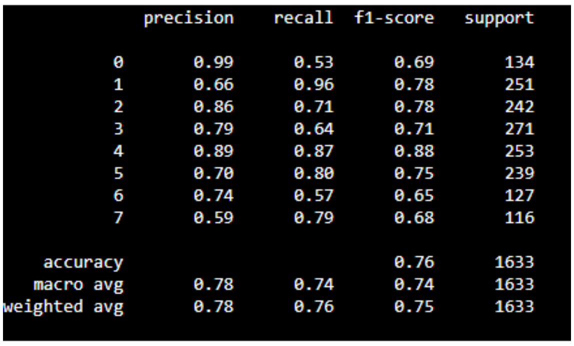
- *Evaluation Metrics for Decision Tree:* The decision tree model are trained on the dataset using feature extraction for each sample. The average precision obtained is 0.78 and recall is 0.78. The f1-score of the model is 0.79 with average support of 1633 and the accuracy of decision tree model is 0.79.



```
              precision    recall  f1-score   support

           0       0.74      0.81      0.77       134
           1       0.87      0.84      0.85       251
           2       0.82      0.71      0.76       242
           3       0.74      0.73      0.73       271
           4       0.83      0.84      0.84       253
           5       0.75      0.84      0.79       239
           6       0.75      0.72      0.74       127
           7       0.73      0.78      0.75       116

    accuracy                           0.79      1633
   macro avg       0.78      0.78      0.78      1633
weighted avg       0.79      0.79      0.78      1633
```

Figure 5. Evaluation metrics for decision tree

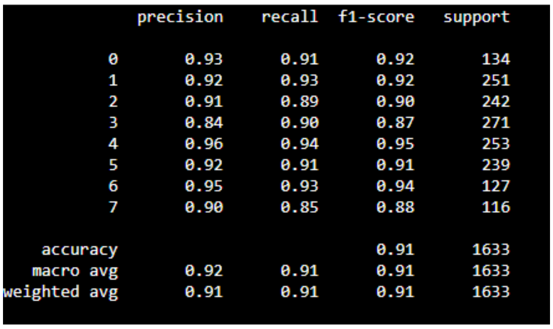
- *Evaluation Metrics for Random Forest:* The Random Forest model are trained on the dataset using feature extraction for each sample. The average precision obtained is 0.78 and recall is 0.74. The f1-score of the model is 0.76 with average support of 1633 and the accuracy of random forest model is 0.76.



```
              precision    recall  f1-score   support

           0       0.99      0.53      0.69       134
           1       0.66      0.96      0.78       251
           2       0.86      0.71      0.78       242
           3       0.79      0.64      0.71       271
           4       0.89      0.87      0.88       253
           5       0.70      0.80      0.75       239
           6       0.74      0.57      0.65       127
           7       0.59      0.79      0.68       116

    accuracy                           0.76      1633
   macro avg       0.78      0.74      0.74      1633
weighted avg       0.78      0.76      0.75      1633
```

Figure 6. Evaluation metrics for Random Forest

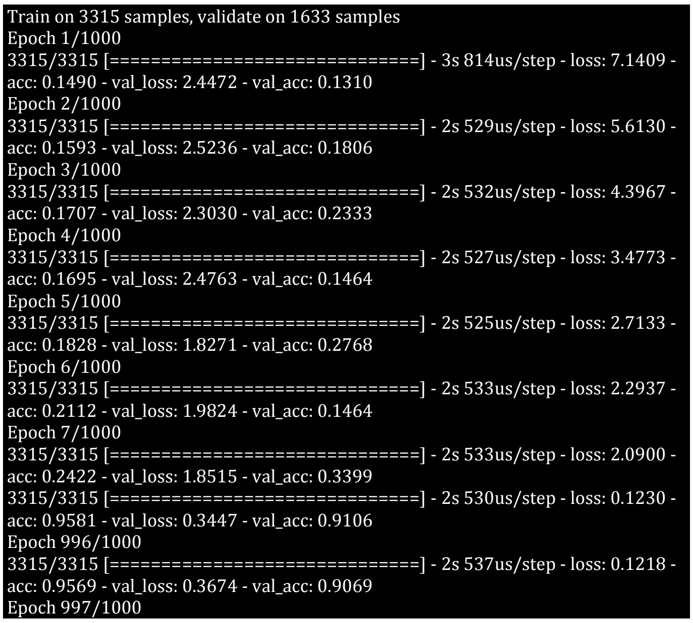
- *Evaluation Metrics for Deep Neural Network:* The Deep Neural network model are trained on the dataset using feature extraction for each sample. The average precision obtained is 0.92 and recall is 0.91. The f1-score of the model is 0.91 with average support of 1633 and the accuracy of random forest model is 0.91.



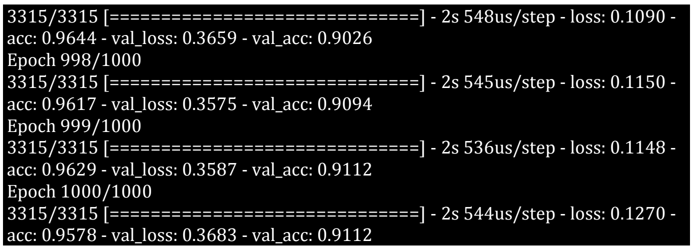
```
              precision    recall  f1-score   support

           0       0.93      0.91      0.92       134
           1       0.92      0.93      0.92       251
           2       0.91      0.89      0.90       242
           3       0.84      0.90      0.87       271
           4       0.96      0.94      0.95       253
           5       0.92      0.91      0.91       239
           6       0.95      0.93      0.94       127
           7       0.90      0.85      0.88       116

    accuracy                           0.91      1633
   macro avg       0.92      0.91      0.91      1633
weighted avg       0.91      0.91      0.91      1633
```

Figure 7. Evaluation metrics for deep neural network

- *Training performed on Deep Neural Network Model:* The deep neural network model train on 3315 samples, validate on 1633 samples with 1000 epoch. The accuracy of DNN model is 0.91.

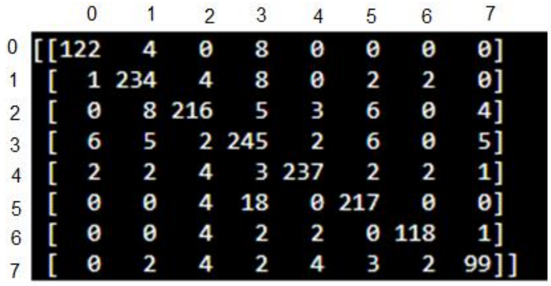


```
Train on 3315 samples, validate on 1633 samples
Epoch 1/1000
3315/3315 [==============================] - 3s 814us/step - loss: 7.1409 -
acc: 0.1490 - val_loss: 2.4472 - val_acc: 0.1310
Epoch 2/1000
3315/3315 [==============================] - 2s 529us/step - loss: 5.6130 -
acc: 0.1593 - val_loss: 2.5236 - val_acc: 0.1806
Epoch 3/1000
3315/3315 [==============================] - 2s 532us/step - loss: 4.3967 -
acc: 0.1707 - val_loss: 2.3030 - val_acc: 0.2333
Epoch 4/1000
3315/3315 [==============================] - 2s 527us/step - loss: 3.4773 -
acc: 0.1695 - val_loss: 2.4763 - val_acc: 0.1464
Epoch 5/1000
3315/3315 [==============================] - 2s 525us/step - loss: 2.7133 -
acc: 0.1828 - val_loss: 1.8271 - val_acc: 0.2768
Epoch 6/1000
3315/3315 [==============================] - 2s 533us/step - loss: 2.2937 -
acc: 0.2112 - val_loss: 1.9824 - val_acc: 0.1464
Epoch 7/1000
3315/3315 [==============================] - 2s 533us/step - loss: 2.0900 -
acc: 0.2422 - val_loss: 1.8515 - val_acc: 0.3399
3315/3315 [==============================] - 2s 530us/step - loss: 0.1230 -
acc: 0.9581 - val_loss: 0.3447 - val_acc: 0.9106
Epoch 996/1000
3315/3315 [==============================] - 2s 537us/step - loss: 0.1218 -
acc: 0.9569 - val_loss: 0.3674 - val_acc: 0.9069
Epoch 997/1000
```



```
3315/3315 [==============================] - 2s 548us/step - loss: 0.1090 -
acc: 0.9644 - val_loss: 0.3659 - val_acc: 0.9026
Epoch 998/1000
3315/3315 [==============================] - 2s 545us/step - loss: 0.1150 -
acc: 0.9617 - val_loss: 0.3575 - val_acc: 0.9094
Epoch 999/1000
3315/3315 [==============================] - 2s 536us/step - loss: 0.1148 -
acc: 0.9629 - val_loss: 0.3587 - val_acc: 0.9112
Epoch 1000/1000
3315/3315 [==============================] - 2s 544us/step - loss: 0.1270 -
acc: 0.9578 - val_loss: 0.3683 - val_acc: 0.9112
```

- *Confusion Metrics of Deep Neural Network Model:* A Confusion matrix is an N x N matrix used for evaluating the performance of a classification model, where N is the number of target classes. The matrix compares the actual target values with those predicted by the machine learning model. This gives us a holistic view of how well our classification model is performing and what kinds of errors it is making.



```
      0    1    2    3    4    5    6    7
0 [[122    4    0    8    0    0    0    0]
1  [  1  234    4    8    0    2    2    0]
2  [  0    8  216    5    3    6    0    4]
3  [  6    5    2  245    2    6    0    5]
4  [  2    2    4    3  237    2    2    1]
5  [  0    0    4   18    0  217    0    0]
6  [  0    0    4    2    2    0  118    1]
7  [  0    2    4    2    4    3    2   99]]
```

Figure 8. Confusion metrics for classification model

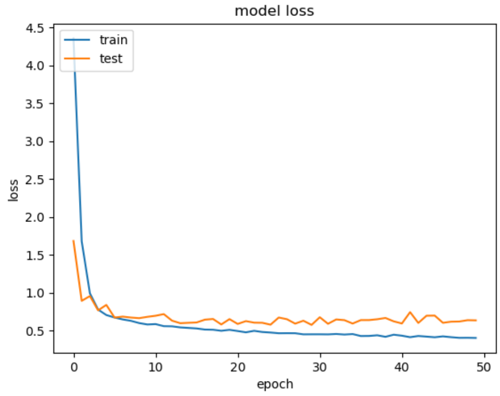


Figure 9. Trend of the cost function of our deep learning model over 1000 epochs.

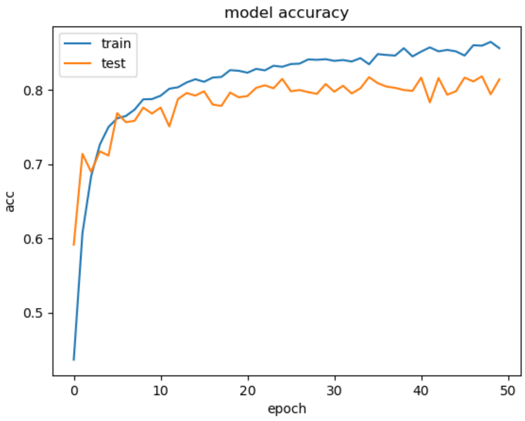


Figure 10. Trend of the accuracy of our deep learning model over 1000 epochs

The authors consider the results encouraging: having a dataset bigger than the RAVDESS available, the MFCC can probably be a valid emotion detection feature. We are sure enough that the same model structure can perform similar results also on audio sound files less structured and collected directly in a real noise environment. The MFCC transformation is always applicable, and using strategies of noise reduction and enough training data; the same model could perform well. As a consequence of this, we are working on experimenting with pieces of dialog directly collected from real users as a future extension of this work.

6. CONCLUSIONS

In this work, we presented an architecture based on deep neural networks for the classification of emotions using audio recordings from the Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS). The model has been trained to classify seven different emotions (neutral, calm, happy, sad, angry, fearful, disgust, surprised) and obtained an overall F1 score of 0.91 with the best performances on the angry class (0.95) and worst on the sad class (0.87). To obtain such result, we extracted the MFCC features (spectrum-of-a-spectrum) from the audio files used for the training. On the above representations of input data, we trained a deep neural network that uses 1D CNNs, max-pooling operations and Dense Layers to estimate the probability of distribution of annotation classes correctly. The approach was tested on the data provided by the RAVDESS & TESS dataset. As baseline for our task, we considered is a random forest classifier we trained on the same dataset achieving an average F1 score of 0.75 over the 8 classes. We have successfully detected the emotions from speech samples and compared three models on basis of various evaluation metrics as described in chapter 4. We can conclude from our research that the DNN performed the good accuracy its time complexity for training is a serious drawback.

We can greatly improve on this drawback by implementing a parallelized model for performing the training of the DNN model. We can also greatly increase the accuracy of the DNN by using more sophisticated layers as described.

This research can be extended to a real world scenario to be put to great use. The trained models can be saved and served as an Application Programme Interface (API) over the web for real time identification of emotions. This can be used as a service. There is a great need of such kind of detection for calls to emergency numbers to confirm intention of the caller. It can also be used in hospitals to gauge the emotional make-up of the caller which can be helpful to build a patient profile. It can also be without a doubt used by marketing professionals to target the users on basis of emotional make-up too.

In conclusion we have successfully achieved the objective of detecting emotions while there are certain drawbacks to the methods implemented in this research all the findings submitted are accurate. After the random forest, we trained a decision tree classifier that achieved an F1 score of 0.78. Our final choice was a deep learning model that obtained a F1 score of 0.91 on the test set.

The good results obtained suggest that such approaches based on deep neural networks are an excellent basis for solving the task. In particular, they are general enough to work in a real application context correctly. Since the result obtained can only be considered a starting point for further extensions, modifications, and improvements of the proposed approach, with the hope that it will be useful for future work in the field.

REFERENCES

[1] DAVIS, S., AND MERMELSTEIN, P. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. IEEE transactions on acoustics, speech, and signal processing 28, 4 (1980), 357–366.

[2] EKMAN, P. Basic emotions. Handbook of cognition and emotion 98, 45-60 (1999), 16.

[3] GOODFELLOW, I., BENGIO, Y., COURVILLE, A., AND BENGIO, Y.

Deep learning, vol. 1. MIT press Cambridge, 2016.

[4] HAYNES, J.-D., AND REES, G. Neuroimaging: decoding mental states from brain activity in humans. Nature Reviews Neuroscience 7, 7 (2006), 523.

[5] HUANG, X., ACERO, A., HON, H.-W., AND FOREWORD BY-REDDY, R. Spoken language processing: A guide to theory, algorithm, and system development. Prentice hall PTR, 2001.

[6] IQBAL, A., AND BARUA, K. A real-time emotion recognition from speech using gradient boosting. In 2019 International Conference on Electrical, Computer and Communication Engineering (ECCE) (2019), IEEE, pp. 1–5.

[7] JANNAT, R., TYNES, I., LIME, L. L., ADORNO, J., AND CANAVAN, S. Ubiquitous emotion recognition using audio and video data. In Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers (2018), ACM, pp. 956–959.

[8] LECUN, Y., BENGIO, Y., ET AL. Convolutional networks for images, speech, and time series. The handbook of brain theory and neural networks 3361, 10 (1995), 1995.

[9] LIVINGSTONE, S. R., AND RUSSO, F. A. The Ryerson audio-visual database of emotional speech and song (ravdess): A dynamic, multi-modal set of facial and vocal expressions in North American English. PloS one 13, 5 (2018), e0196391.

[10] LOGAN, B., ET AL. Mel frequency cepstral coefficients for music modeling. In ISMIR (2000), vol. 270, pp. 1–11.

[11] MUDA, L., BEGAM, M., AND ELAMVAZUTHI, I. Voice recognition algorithms using mel frequency cepstral coefficient (mfcc) and dynamic time warping (dtw) techniques. arXiv preprint arXiv:1003.4083 (2010).

[12] NAIR, V., AND HINTON, G. E. Rectified linear units improve restricted boltzmann machines. In Proceedings of the 27th international conference on machine learning (ICML-10) (2010), pp. 807–814.

[13] PEDREGOSA, F., VAROQUAUX, G., GRAMFORT, A., MICHEL, V., THIRION, B., GRISEL, O., BLONDEL, M., PRETTENHOFER, P., WEISS, R., DUBOURG, V., VANDERPLAS, J., PASSOS, A., COURNA-PEAU, D., BRUCHER, M., PERROT, M., AND DUCHESNAY, E. Scikit-earn: Machine learning in Python. Journal of Machine Learning Research 12 (2011), 2825–2830.

[14] PLATT, J. C., CRISTIANINI, N., AND SHAWE-TAYLOR, J. Large margin dags for multiclass classification. In Advances in Neural Information Processing Systems 12, S. A. Solla, T. K. Leen, and K. Muller, Eds. MIT Press, 2000, pp. 547–553.

[15] POLIGNANO, M., DE GEMMIS, M., NARDUCCI, F., AND SEMERARO, G. Do you feel blue? detection of negative feeling from social media, 2017.

[16] ZHANG, B., ESSL, G., AND PROVOST, E. M. Recognizing emotion from singing and speaking using shared models. In 2015 International Conference on Affective Computing and Intelligent Interaction (ACII) (2015), IEEE, pp. 139–145.