MANTIS - A Voice Assistant Predicting Real Time Emotions from Audio and Video

Chaitanya Virmani
Manav Rachna University
2K19CSUN04009

Mayank Saxena
Manav Rachna University
2K19CSUN04015

1. Introduction

We know today's world is running at a very fast pace. People try their best to compete with one another and contribute to society. But between all these they get very less time for themselves. We go though many things, many emotions daily. Stabilizing our emotions has become a big challenge. People don't know how to deal with anger, sadness, failure and many other emotions. This sometimes becomes the reason they took up steps that they shouldn't. In addition, in maintain a healthy lifestyle emotional stability is very important. This motivated us to build a personal psychiatrist- Mantis, that will not only recognize your emotions but also give you advices what you should do and help you cope up with every situation.

MANTIS – is a virtual assistant that will interact with human in real time and predict his/her emotion by capturing image and audio. Based on the emotion it will give advices to that person and tell what they should do to deal with it and maintain the emotional stability.

2. Related Work

A lot of work has already been done in this area. Emotion recognition is predicted by audio, video, expressions and even by words. Classical machine learning algorithms, such as convolution neural networks, support vector machines (SVMs), and classifier methods, have been employed in emotion recognition problems. Various neural network based architecture have also been introduced by the researchers for the improved predictions. An initial study utilized deep neural networks (DNNs) to extract high-level features from raw audio data and demonstrated its effectiveness in speech emotion recognition.

Researcher investigated transfer learning methods, leveraging external data from related domains. As emotional dialogue is composed of sound and spoken content, researchers have also investigated the combination of acoustic features and language information. However, very few of these studies have utilized information from speech signals and facial expression sequences simultaneously in an end-to-end learning neural network-based model to classify emotions.

3. Dataset and Features

3.1 Datasets used

3.1.1 FER 2013(Facial Expression Recognition)

We used FER-2013 dataset to train our model for emotion recognition from face. The FER-2013 dataset consists of 28,000 labelled images in the training set, 3,500 labelled images in the development set, and 3,500 images in the test set. Each image in FER-2013 is labelled as one of seven emotions: happy, sad, angry, afraid, surprise, disgust, and neutral, with happy being the most prevalent emotion, providing a baseline for random guessing of 24.4%. The images in FER-2013 consist of both posed and unposed headshots, which are in grayscale and 48x48 pixels. The FER-2013 dataset was created by gathering the results of a Google image search.

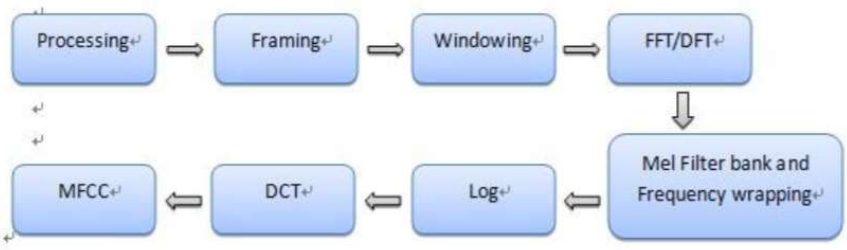
3.1.2 Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS)

The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) contains 7356 files(24.8GB). But we have only used speech files. The database contains 24 professional actors (12 females, 12 males), vocalizing two lexically-matched statements in a neutral North American accent. Speech includes calm, happy, sad, angry, fearful, surprise, and disgust expressions. All conditions are available in three modality formats: Audio-only (16bit, 48kHz .wav), Audio-Video (720p H.264, AAC 48kHz, .mp4), and Video-only (no sound). But we have used audio-only files, reducing the dataset to 1.1GB.

3.2 Features Extracted

We have extracted MFCC features from audio files. The MFCC feature extraction technique basically includes windowing the signal, applying the DFT, taking the log of the magnitude, and then warping the frequencies on a Mel scale, followed by applying the inverse DCT.

For predicting emotions from face, we already have image dataset (RAVDESS), that contains 48x48 grey scale images. So we haven't done much feature extraction on image files.



4 Methods

4.1 Methods used for FER (Facial Emotion Recognition)

4.1.1 CNN

CNN is a neural network that consists of several layers of different types, including convolutional, max pooling, ReLU activation, dropout, dense/fully connected, and softmax. Each convolutional layer is a set of learnable 2D filters, which are applied to input data by the 2D cross-correlation operation. In max pooling layers, data is downsampled by a set factor n by choosing only the max value in each nxn square topropa gate forward. The ReLU activation function is a unit ramp function f(x) = max (x; 0) that allows for non-linearlity in the network. In dense layers, input data is flattened into 1D vectors, multiplied by a matrix of learnable weights, and added with a learnable bias. Dropout removes a percentage of activations to help prevent overfitting. Finally, the softmax layer computes the class probabilities for the data.

In our CNN model, each convolutional layer uses a ReLU activation function. These convolutional and max-pooling layers are followed by two dropout layers with a keep-probability of 0.25 and dense layer, and another dropout layer with keep-probability of 0.5. Finally, cross-entropy loss is computed on the outputs of the softmax as activation function.

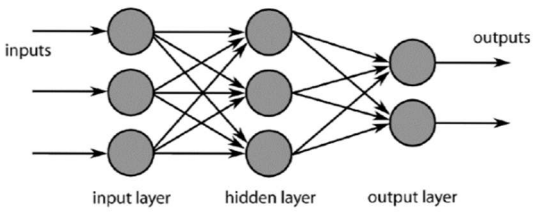
Softmax Function: $$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

$\vec{z}$	The input vector to the softmax function, made up of (z0, … zK)
z_i	All the zi values are the elements of the input vector to the softmax function, and they can take any real value, positive, zero or negative. For example a neural network could have output a vector such as (-0.62, 8.12, 2.53), which is not a valid probability distribution, hence why the softmax would be necessary.
e^{z_i}	The standard exponential function is applied to each element of the input vector. This gives a positive value above 0, which will be very small if the input was negative, and very large if the input was large. However, it is still not fixed in the range (0, 1) which is what is required of a probability.

$\sum_{j=1}^{K} e^{z_j}$	The term on the bottom of the formula is the normalization term. It ensures that all the output values of the function will sum to 1 and each be in the range (0, 1), thus constituting a valid probability distribution.
K	The number of classes in the multi-class classifier.

Hidden layers in the model:

Input Layer	(48, 48, 1)
Conv2d	(None, 46, 46, 32)
Conv2d	(None, 44, 44, 64)
Max-pooling2d	(None, 22, 22, 64)
Dropout	(None, 22, 22, 64)
Conv2d	(None, 20, 20, 128)
Maxpooling2d	(None, 10, 10, 128)
Conv2d	(None, 8, 8, 128)
Max-pooling2d	(None, 4, 4, 128)
Dropout	(None, 4, 4, 128)
Flatten	(None, 2048)
Dense	(None, 1024)
Dropout	(None, 1024)
Dense	(None, 7)



4.2 Methods used for SER (Speech Emotion Recognition)

4.2.1 CNN

In our model we have used 3 convolution 1D layers each followed with a max-pooling layer and then a dropout layer with a keep-probability of 0.2, then again a convolution 1D

layer followed with a max-pooling layer. Each convolution layer has ReLU as activation function. Then we have used flatten layer to flatten our data into 1D Vectors. This layer is followed by a dense layer with ReLU as activation function and then a dropout layer with a keep-probability of 0.3. At last a dense layer is used to compute cross-entropy loss on the outputs of the softmax as activation function.

Hidden layers in the model:

Input Layer	(40,1)
Conv1d	(None, 40, 256)
Max-pooling	(None, 20, 256)
Conv1d	(None, 20, 256)
Max-pooling	(None, 10, 256)
Conv1d	(None, 10, 128)
Max-pooling	(None, 5, 128)
Dropout	(None, 5, 128)
Conv1d	(None,5, 64)
Max-pooling	(None,3,64)
Flatten	(None,192)
Dense	(None,32)
Dropout	(None, 32)
Dense	(None, 8)

4.2.2 RNN

In RNN we have used LSTM layer. Long short-term memory (LSTM) is an artificial recurrent neural network (RNN) architecture used in the field of deep learning. Unlike standard feedforward neural networks, LSTM has feedback connections. It can not only process single data points (such as images), but also entire sequences of data (such as speech or video).

The LSTM layer is followed with 2 dense and 2 dropout layers with each dropout layer having a keep-probability of 0.4. Activation function used with each layer is reLU. Finally, a dense layer is used to compute cross-entropy loss on the outputs of the softmax as activation function.

Hidden layers in the model:

Input Layer	(40,1)
LSTM	(None,128)
Dense	(None, 64)
Dropout	(None, 64)
Activation(ReLU)	(None, 64)
Dense	(None, 32)
Dropout	(None, 32)
Activation(ReLU)	(None, 32)
Dense	(None, 8)
Activation(softmax)	(None, 8)

4.2.3 MLP Classifier

A multilayer perceptron (MLP) is a class of feedforward artificial neural network (ANN). The term MLP is used ambiguously, sometimes loosely to any feedforward ANN, sometimes strictly to refer to networks composed of multiple layers of perceptron (with threshold activation). Multilayer perceptron is sometimes colloquially referred to as "vanilla" neural networks, especially when they have a single hidden layer.

In our model we have used 1 hidden layer with 300 neurons, learning rate is 0.001 by default and set to adaptive. Activation function used is ReLU.

5 Experiments

Our models are implemented using keras. Feature extraction was relied on librosa library for audio files and openCV for image files. All the models are trained using Adam as optimizer. Learning rate was set to adaptive. Accuracy was improved by changing the

neurons in each layer and adjusting the learning rate. Keep-probability in dropout layers were adjusted accordingly.

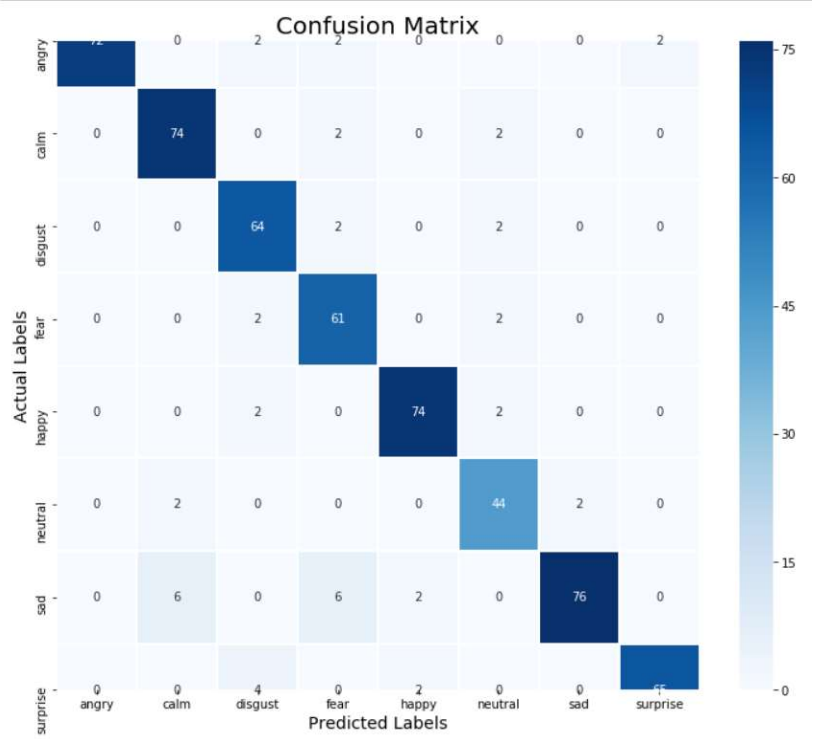
We have increased hidden layers in CNN model to improve accuracy. Max-pooling layers were added to reduce overfitting.

6 Results and discussion

We have used accuracy as metric to evaluate our model. For each model accuracy was calculated separately on training and testing datasets. Our dataset for both audio and image file was balanced.
The accuracy calculated for each model is:

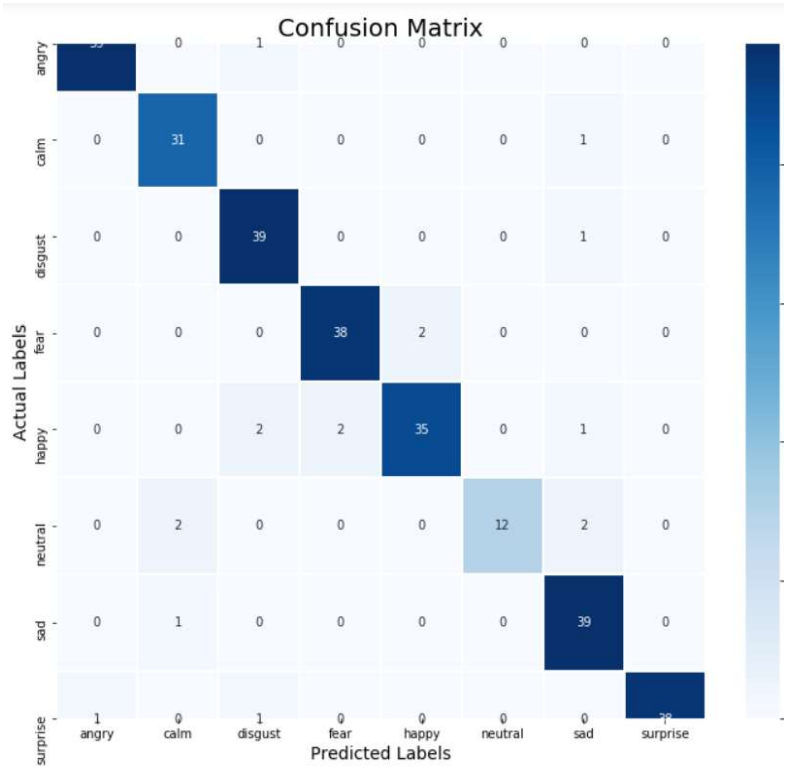
Model Used	Accuracy on training set	Accuracy on test Set
RNN	96.61	94.09
CNN	98%	92.01%
MLP Classifier	100%	93.75%
CNN(FER)	96%	63.01%

The confusion matrix and classification report of various SER models are here



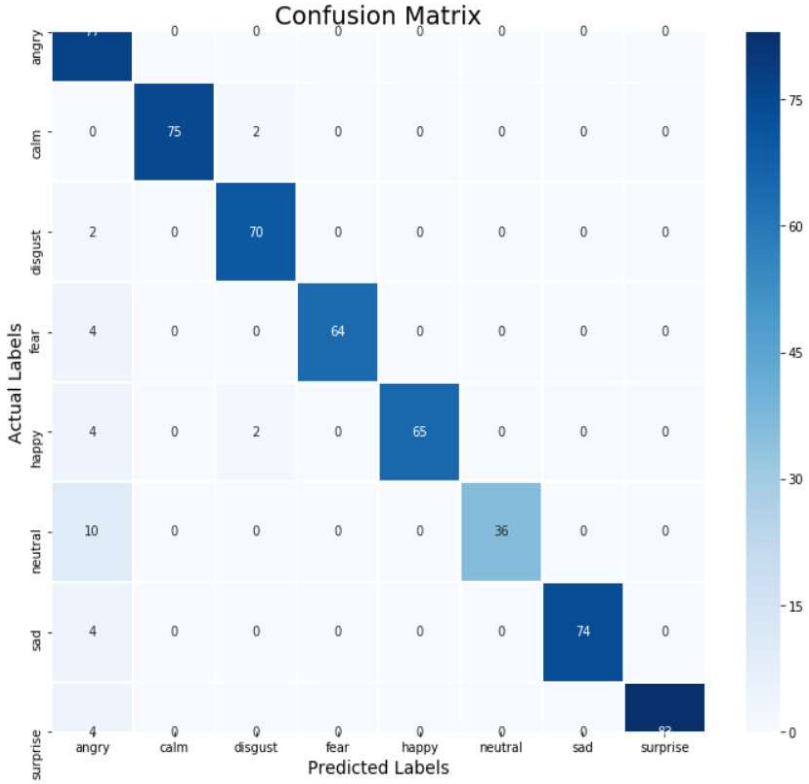
	precision	recall	f1-score	support
angry	1.00	0.92	0.96	78
calm	0.90	0.95	0.92	78
disgust	0.86	0.94	0.90	68
fear	0.84	0.94	0.88	65
happy	0.95	0.95	0.95	78
neutral	0.85	0.92	0.88	48
sad	0.97	0.84	0.90	90
surprise	0.97	0.92	0.94	71
accuracy			0.92	576
macro avg	0.92	0.92	0.92	576
weighted avg	0.92	0.92	0.92	576

CNN Model



	precision	recall	f1-score	support
angry	0.87	1.00	0.93	40
calm	0.89	1.00	0.94	32
disgust	1.00	0.88	0.93	40
fear	1.00	0.93	0.96	40
happy	0.94	0.85	0.89	40
neutral	0.86	0.75	0.80	16
sad	0.97	0.97	0.97	40
surprise	0.91	1.00	0.95	40
accuracy			0.93	288
macro avg	0.93	0.92	0.92	288
weighted avg	0.94	0.93	0.93	288

RNN Model



	precision	recall	f1-score	support
angry	0.73	1.00	0.85	77
calm	1.00	0.97	0.99	77
disgust	0.95	0.97	0.96	72
fear	1.00	0.94	0.97	68
happy	1.00	0.92	0.96	71
neutral	1.00	0.78	0.88	46
sad	1.00	0.95	0.97	78
surprise	1.00	0.95	0.98	87
accuracy			0.94	576
macro avg	0.96	0.94	0.94	576
weighted avg	0.96	0.94	0.95	576

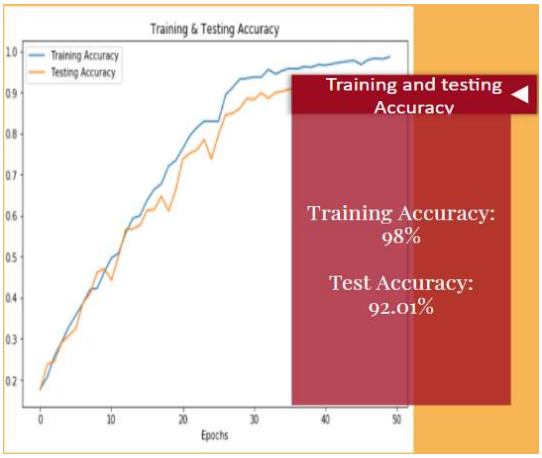
MLP Classifier

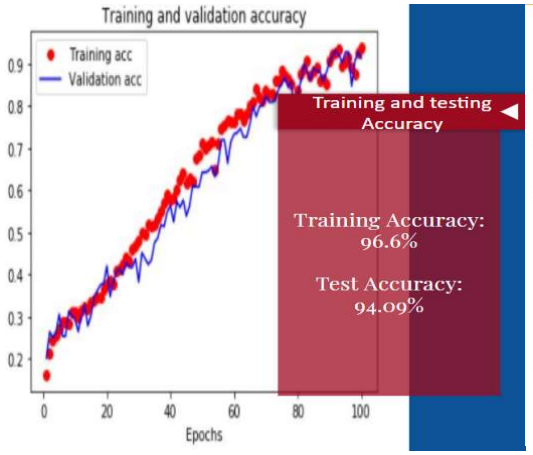
Finalising the two models, one for facial emotion recognition and one for speech emotion recognition, then we have built a voice assistant using pyttsx3 library that converts text to speech. PyAudio is used to record audio as an input from user to predict emotion based on speech. We have also used SpeechRecognition library to recognise what user says and based upon that, our voice assistant Mantis can take actions.

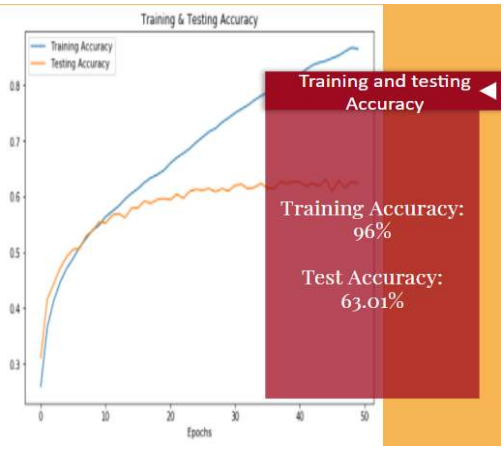
After it has predicted emotion, it asks the user whether the predicted emotion is right. If user agrees, gives advices what user should do based on his/ her choice. When user disagrees with the predicted emotion, it asks the user what he or she is feeling, and gives advice according to that emotion if the user wants to hear.

7 Conclusion and Future Work

We explored the task of emotion detection through voice as well as face in real time. With accuracies as follows:







We can see from classification report in Results and discussions [6] that all of our models are performing almost same on training data. CNN model is predicting angry, calm, disgust, happy, sad, surprise with high f1 scores whereas RNN is predicting angry, calm, fear, sad, surprise with maximum f1 scores and MLP classifier is predicting calm, disgust, fear, happy, sad, surprise with high f1 scores.

We have used RNN model for predicting emotions through speech because it is giving maximum accuracy on test data.

Further, we can use our model to help people get sound sleep and in meditation. This can also be used in cars for detecting driver's emotion and can help in avoiding road accidents.

References

[1] Bj¨orn Schuller, Gerhard Rigoll, and Manfred Lang, “Hidden markov model-based speech emotion recognition,” in Multimedia and Expo, 2003. ICME’03. Proceedings. 2003 International Conference on. IEEE, 2003, vol. 1, pp. I–401.

[2] Chi-Chun Lee, Emily Mower, Carlos Busso, Sungbok Lee, and Shrikanth Narayanan, “Emotion recognition using a hierarchical binary decision tree approach,” Communication, vol. 53, no. 9-10, pp. 1162– 1171, 2011.

[3] Kun Han, Dong Yu, and Ivan Tashev, “Speech emotion recognition using deep neural network and extreme learning machine,” in Fifteenth Annual Conference of the International Speech Communication Association, 2014.

[4] Dario Bertero and Pascale Fung, “A first look into a convolutional neural network for speech emotion detection,” in Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on. IEEE, 2017, pp. 5115–5119.

[5] Abdul Malik Badshah, Jamil Ahmad, Nasir Rahim, and Sung Wook Baik, “Speech emotion recognition from spectrograms with deep convolutional neural network,” in Platform Technology and Service (PlatCon), 2017 International Conference on. IEEE, 2017, pp. 1–5.

[6] Zakaria Aldeneh and Emily Mower Provost, “Using regional saliency for speech emotion recognition,” in Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on. IEEE, 2017, pp. 2741–2745

[7] Aharon Satt, Shai Rozenberg, and Ron Hoory, “Efficient emotion recognition from speech using deep learning on spectrograms,”

[8] Dong Yu and Li Deng, AUTOMATIC SPEECH RECOGNITION., Springer, 2016.

[9] Seyedmahdad Mirsamadi, Emad Barsoum, and Cha Zhang, “Automatic speech emotion recognition using recurrent neural networks with local attention,” in Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on. IEEE, 2017, pp. 2227–2231.

[10] Hao Zhou, Minlie Huang, Tianyang Zhang, Xiaoyan Zhu, and Bing Liu, “Emotional chatting machine: Emotional conversation generation with internal and external memory,” 2018.

[11] Chenyang Huang, Osmar Zaiane, Amine Trabelsi, and Nouha Dziri, “Automatic dialogue generation with expressed emotions,” in Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2018, vol. 2, pp. 49–54.

[12] Carlos Busso, Murtaza Bulut, and Shrikanth Narayanan, “Toward effective automatic recognition systems of emotion in speech,” Social