INDIAN INSTITUTE OF INFORMATION TECHNOLOGY ALLAHABAD
(A Centre of Excellence in Information Technology) Deoghat, Jhalwa, Allahabad – 211 012 (U.P.),
India Ph. 0532-2922008, 2922216, Fax: 0532-2430006,
Web: www.iiita.ac.in, E-mail: dr.e@iiita.ac.in

Semester VIII – End Semester Project Report

Project Title

Multi-Modal Emotion Classification using User's Speech & Facial Expressions

Under the Guidance of:

Prof. U.S. Tiwari

Submission By:
Vaibhav Srivastava (IIT2013027)
Himanshu Tuteja (IIT2013038)
Anirudh Gupta (IIT2013117)

CERTIFICATION

This is to certify that the group of below mentioned students has successfully completed the project work titled **"Multi-Modal Emotion Classification using User's Speech & Facial Expressions"** as the VIII semester Major-Project prescribed by the Indian Institute of Information Technology, Allahabad.

This project is the record of authentic work carried out during the academic year (Jan – April) 2017.

.

Vaibhav Srivastava (IIT2013027)
Himanshu Tuteja (IIT2013038)
Anirudh Gupta (IIT2013117)

ACKNOWLEDGEMENT

The present work took a lot of our efforts. However, the work carried out would not have been possible without the support and effort of many people around. We would like to thank all of them for their throughout help and support.

We are highly indebted to **Prof. U.S. Tiwari** for his expert guidance and throughout during the entire course of the project. We would like to thank him for providing the required information and assistance to complete the project.

The project would not have been possible without the kind cooperation and support **Mr. Sudhakar Mishra (RS163)**. Our special thanks also goes to our colleague and friends in carrying out the task and also to the people who have willingly helped us out with their abilities.

Table of Contents

1. Abstract

One of challenging task today is making systems that can behave as humans and improve the human computer interaction. Extracting emotions from the speech signals is one such problem and an effective solution can help in providing some fruitful results in the coming future. This problem is not as easy as one thinks, it is a bit more difficult when compared with speech or speaker recognition problem. Speech signals are one of the main communication medium. Analysing the acoustic differences while occurrence of similar thing under different emotional states is one of the major problem faced by researchers. Also adjusting the sound signals to match the static learning architecture is also a crucial task. The present work aims at exploring the various dependencies the nature of utterances have with the emotional state of humans. The human nervous system is directly influenced by the emotions and so does the heart rate. One interesting note that can be observed is that the speech signals can be treated as a representative of user's heart rate.

2. Project Title

MULTI-MODAL EMOTION CLASSIFICATION USING USER'S SPEECH & FACIAL EXPRESSION

3. Introduction

3.1 The Problem

The features that distinguish human beings from machines and robots are emotions. Human beings express their emotions in every moment using various different modalities like actions, facial expressions, speech etc. For machines, analysing, recognising and generating emotions can play a crucial role in reducing the gap between the mankind and artificial intelligence. Emotions play an essential role in decision making, and emotions influence rational thinking of a person and therefore should be part of interactive technologies. The number of human communication channels is reduced in human machine interaction which makes communication more difficult. This encourages the need of a single system that can detect the human emotions using a combination of different modalities and can help in developing a wide range of products which can provide a "state of the art" experience to the user.

One of a huge challenge is developing computers that can effectively simulate human interaction. A lot of research has been done in the past years in this field. This motivates us to develop a system which can efficiently determine the mental state of the user by studying the speech and facial expressions of the user.

3.2 The Solution

We developed a solution in this report for the problem in hand. We propose a system which classifies the emotion of the person based on 2 modalities, speech and facial expressions. The system would be able to give the emotions using various machine learning techniques in real time. Different techniques have different evaluating parameters and different accuracies. Our goal is to find the best suitable method that can be used as the solution to the above problem.

Any learning algorithm that uses features extracted from data will always be upper-bounded in its accuracy by the expressiveness of the features. As a result, a motivation for deep learning based approaches is that learning algorithms can learn, or design features much better than humans can.

4. Project Objectives

By doing this project, the followings objectives are achieved:

a) Development of separate modules to detect the emotion of the person using Facial Expression and Speech Data
    i. **Module 1:** Emotion Classification using Facial Expressions
    ii. **Module 2:** Emotion Classification using Speech Data
b) Fusion of the results of speech and facial expression classifier to enhance the accuracy.
c) Using effective machine learning algorithms to achieve the above mentioned task.
d) Comparing the accuracies obtained by different methodologies and choosing the best one.

5. Scope of Project Work

The scope of work will include the followings:

a) Such an application could be really helpful in **call centres**, where conversations with the customers could be studied, which can further help in improving service quality.

b) Applications like **E-teaching, storytelling** etc. could be more realistic, if they can adjust to the emotional states of the students.

c) In **Home Automation Systems**, emotional sensing ability could help in avoiding the risks of depression by cheering the mood of the subject by playing music, controlling the lights or by calling the loved ones.

d) Such an application could be a big improvement in **human machine interaction**.

e) Inside cars, an application of emotion recognition could examine the mental state of the driver and could help in **avoiding the chances of accidents** as the mental state of the driver is very likely to affect the driving.

f) Such an application could be installed in an **airplane cockpit, where emergency situations** could be detected by examining the stress levels in the speech of the pilots.

6. Literature Survey

The basic task required to accomplish the solution of speech emotion recognition system is understanding the concepts of machine learning and convolution neural networks (CNN) with LSTM and comparing their results to obtain the best possible solution, the one with the highest accuracy

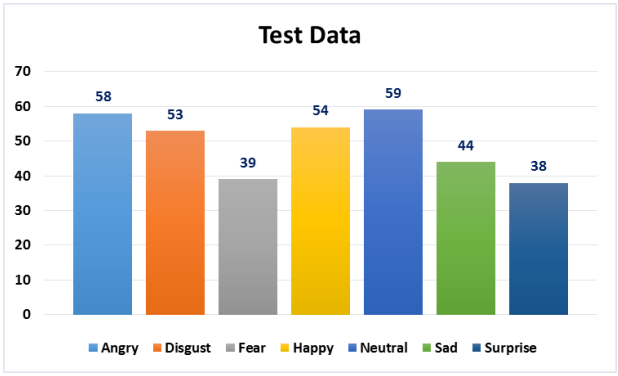
6.1 Data Set [1] [14]

There are a lot of free speech databases available in different languages for classifying human emotions using speech. We first used the online available datasets to train our model, but found the result were not that fruitful as the dataset available for English language was not sufficient. For our final model, the following datasets are used.

- **The Surrey Audio-Visual Expressed Emotion (SAVEE) [1] dataset:** It is a pre-processed dataset and consists of 480 British English utterances recorded by 4 male actors portraying 7 emotions—**happy, angry, sad, disgusted, fearful, surprised, and neutral.** The files are on average 3 seconds long. The files are in .wav format. The database also contains video files with facial expression and speech modulations which can used to test the final system.
- **IIITA Dataset:** Since the SAVEE dataset was not sufficient to train the model and produce effective results, we recorded our own data consisting of 853 audio clips by **82 actors (11 female, 71 male)** expressing the 7 seven emotions. The data was recorded with "Zoom H6" device in proper condition to ensure good quality audio clips without much noise.

Fig. Recording of Dataset

All the audio clips are captured at a frequency of 44100 Hz. The length of each audio file is around 2.0 ~ 3.0 seconds and are stored in wave (.wav) format. The total number of audio clips are 1333, which are further divided into 988 for training and 345 for validating the model.



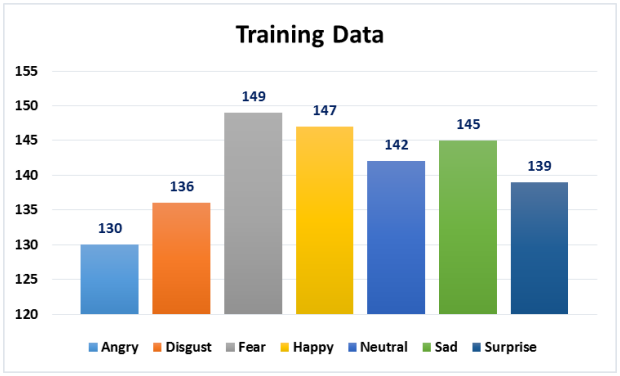
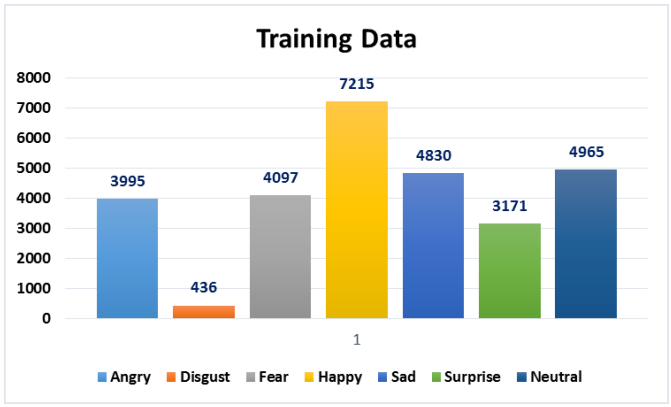


Fig. Speech Dataset

- **Image Dataset:** For facial expression classification we used a dataset provided by FER2013[1] website, which consists of about 32298 well-structured 48 × 48 pixel gray-scale images of faces. The images are processed in such a way that the faces are almost centred and each face occupies about the same amount of space in each image. Each image has to be categorized into one of the seven classes that express different facial emotions.



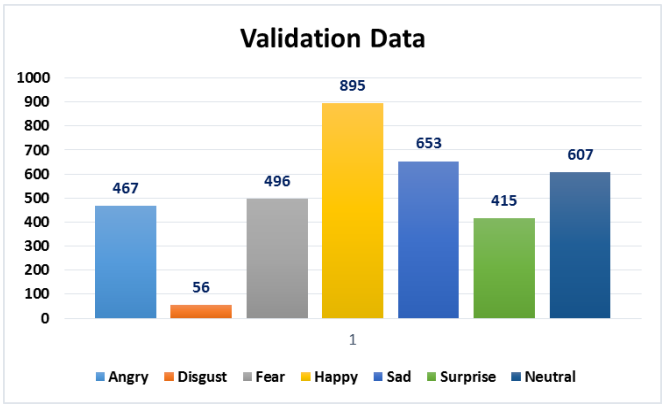


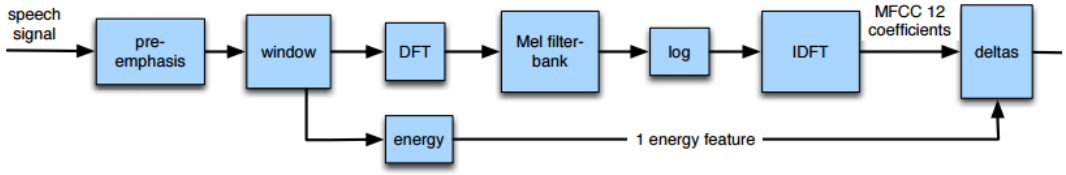
Fig. Image Dataset

6.2 Python[2]

Python is a high-level, interpreted, dynamic and general purpose language which allows a greater readability to the user, as its syntax allows user to use fewer lines to express the concepts. Also it provides constructs to enable programs a small as well as a large scale. Python supports object oriented, procedural, functional and imperative programming paradigms. It also has a very large and comprehensive standard library.

6.3 Mel Frequency Cepstral Coefficients (MFCC)[3][4]

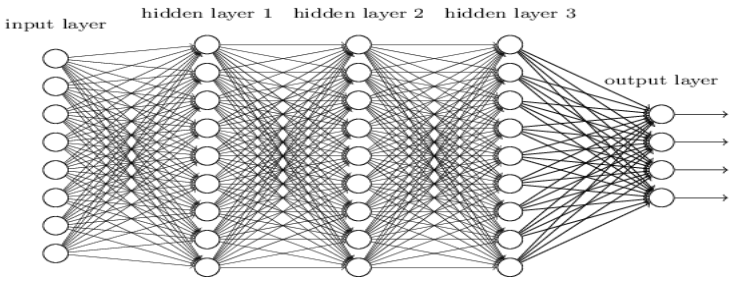
Mel-frequency cepstral coefficients (MFCCs) are a parametric representation of the speech signal that is commonly used in automatic speech recognition, but they have proved to be successful for other purposes as well, among them speaker identification and emotion recognition. MFCCs are calculated by applying a Mel-scale filter bank to the Fourier transform of a windowed signal. Subsequently, a DCT (discrete cosine transform) transforms the logarithmised spectrum into a cepstrum. The MFCCs are then the amplitudes of the cepstrum. Usually, only the first 12 coefficients are used. Through the mapping onto the Mel-scale, which is an adaptation of the Hertz-scale for frequency to the human sense of hearing, MFCCs enable a signal representation that is closer to human perception. MFCCs filter out pitch and other influences in speech that are not linguistically relevant, hence they are very suitable for speech recognition.



6.4 Convolution Neural Network [5][6][7]

Convolutional neural networks (CNNs) are widely used in the tasks of pattern and image recognition because of their superiority over other techniques. CNN resembles neural network. As we know neural network are made up of neurons having some learnable weights and biases. Every neuron in neural network takes some input followed by dot product operation and then follows it with some non-linearity if required. CNN takes raw image pixels as input and outputs the scores of different classes and the last fully connected layer of CNN still have the loss functions like softmax and all the functionalities of neural network can still be incorporated into the CNN.

The basic need of CNN aroused for image recognition problems as in case of images the no of parameters in input layer become large and in order to make the recognizing system efficient the number of hidden layers in the neural network are also large, due to which the effect to the weights of initial hidden layers is not much during back propagation. This increases the number of iterations needed to adjust the weights in order to obtain good accuracy from the system, thereby increasing the computation power.



For example an input image of size 300 x 300 pixels will require 90000 neurons in the input layer. Adjusting weights for such large number of input neurons will require large number of hidden layers and also good accuracy can't be guaranteed.

CNN takes into consideration spatial information in an image. It extracts the important features and trains the system accordingly.

- **Input Layer** :-

  This layer holds the raw pixel values of the image e.g. in this case an image of width and height 48 and 48, and with the three color channels which are R, G and B it would have dimensions 48 x 48 x 3.

- **CONV Layer** :-

  In this layer, each neuron performs a dot product between their weights and their local receptive fields. This may result in volume such as [48x48x12] if we have decided to employ 12 filters.

- **RELU Layer** :-

  This layer performs an element wise activation function, such as the **max(0,x)** which is used to do the threshold at zero. This removes negative intensities while keeping the volume unchanged [48x48x12].

- **POOL layer** :-

  This layer will do a down sampling along the spatial dimensions (width, height), thus resulting a volume as [24x24x12] in case if we are using a 2 x 2 pooling filter. Eg. A 2 x 2 max pool filter.

- **FC Layer**

  This fully-connected layer computes the class scores, resulting in a volume of size [1x1x7], where each number corresponds to a particular class score, among the 7 possible categories. As the name implies, each neuron in this layer will be connected to all the neurons in the previous volume.

The filters in the CONV layer are simple edge detection filters at different angles (horizontal, vertical, +45 degree i.e. diagonals, etc.) as the edges carry the useful information in the image. We can multiple combinations of CONV, RELU and POOL layer in between the Input and the FC layer to try out different architectures in order to obtain best possible results.

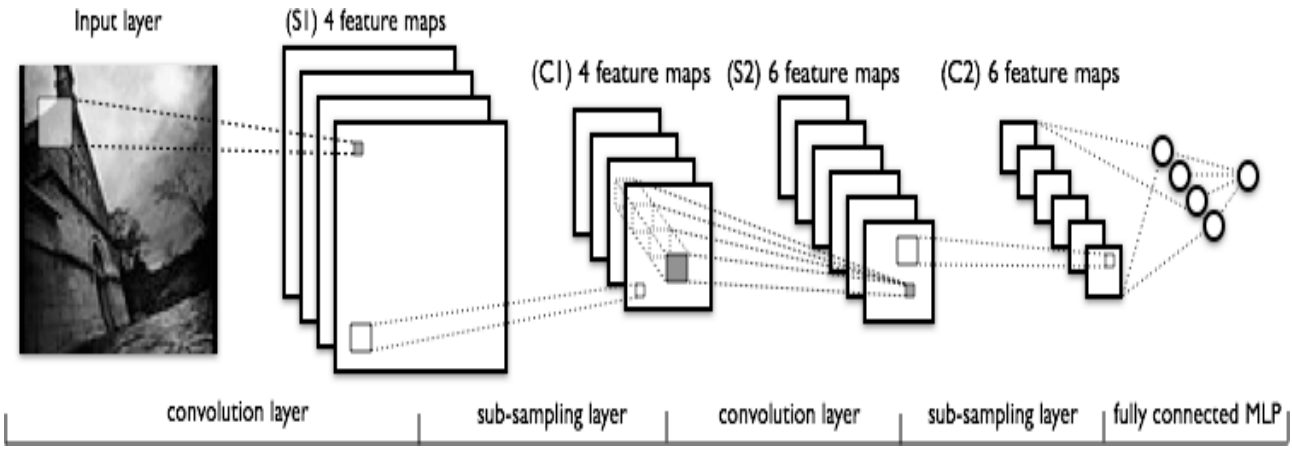


Fig. Basic CNN Architecture

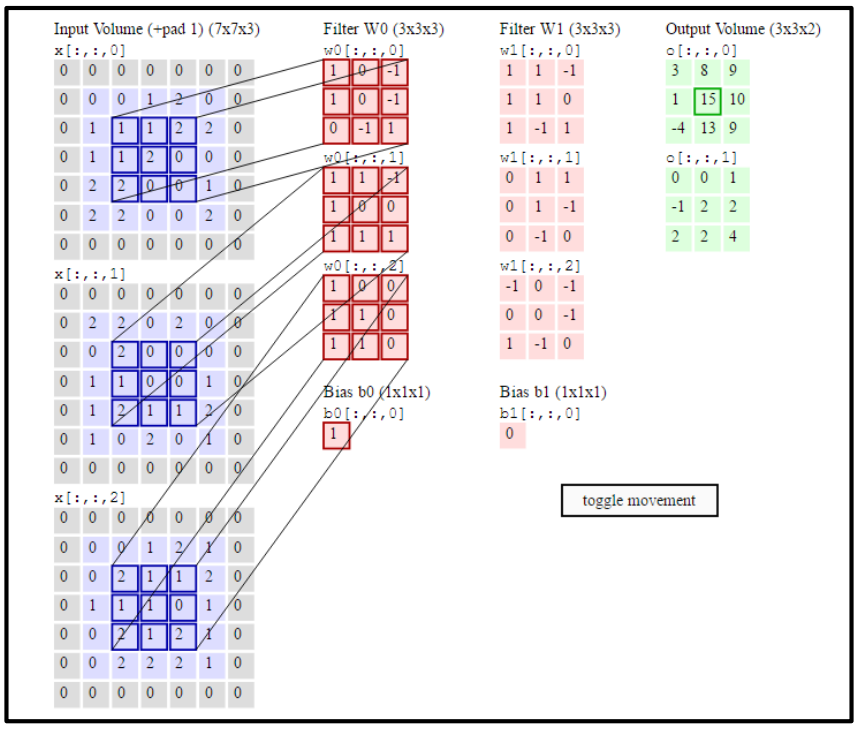
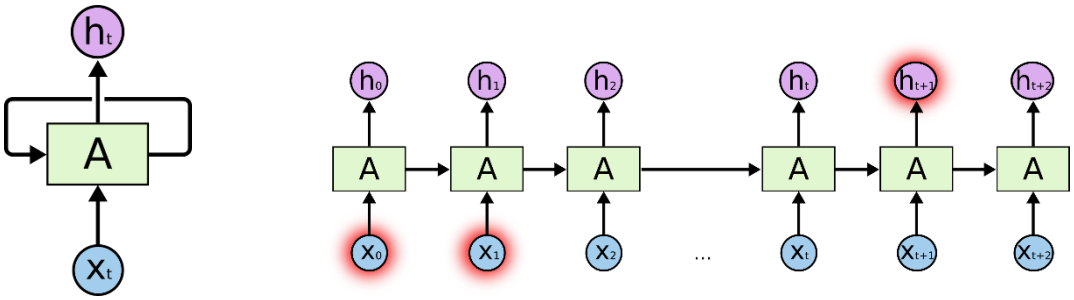


Fig: The spatial filters are applied on the input image (CONV layer)

6.5 Long Short Term Memory (LSTM)[8][13]:

LSTM are like Recurrent Neural Network with some additional features and modification. RNN are neural networks with loop in them, allowing the information to persist. RNN are good to use when relation exist near only i.e. when information is presented and where it is to be used has no or very less gap but if gap increases then RNN is not able to handle such long term dependencies. In order to overcome this, LSTM is used.

LSTM does not struggle to remember information for long periods of time, it is their default behaviour. They have ability to add or remove information which is regulated through gates. Gates in LSTM, consist of a point-wise multiplicative operation and a sigmoid neural net layer. The output which is produced by sigmoid layer or gates is a number between zero and one, which defines the amount of information to be considered.



7. Other Related Work

7.1 I. R. Murray and J. L. Arnott, "*Toward the simulation of emotion in synthetic speech: A review of the literature on human vocal emotion*" [9]

This is an often cited review by Murray and Arnott for review literature on emotions in speech and is referred in a number of studies which have identified several acoustic correlations of emotions. The summary of their review is displayed in Table below which shows prosody and voice quality to be most important to distinguish between emotions according to human perception. In particular, pitch and intensity seem to be correlated to activation, so that high pitch and intensity values imply high, low pitch and intensity values low activation.

Emotion	Pitch	Intensity	Speaking rate	Voice quality
Anger	higher mean wider range abrupt changes	higher	slightly faster	breathy chest tone
Joy	higher mean wider range	higher	faster or slower	breathy blaring
Sadness	lower mean narrower range	lower	slower	resonant
Fear	higher mean wider range	normal	faster	irregular voicing
Disgust	lower mean wider range	lower	slower	grumbled chest tone

Table 1: Variations of acoustic variables observed in emotional speech

7.2 "Convoluted Feelings" Convolutional and recurrent nets for detecting emotion from audio data [10]

This paper aims in classifying emotions from speech data by using Convolutional Neural Network. The result of this paper shows that they can achieve an accuracy of 50% for 7-class classification using CNN extracted features and the accuracy can further be increased by using LSTM with CNN in order to gain context over time, but no proper implementation or insights have been provided to study about these architecture.

7.3 Unsupervised feature learning for audio classification using convolutional deep belief networks [11]

In this paper, Convolutional Deep Belief Network is applied to unlabelled speech data and the features which are learned are used for classification purposes and this paper also shows that features learned from this model outperform other baseline features such as MFCC because they are hand-tailored to the audio data.

7.4 Convolutional Neural Networks for Facial Expression Recognition [12]

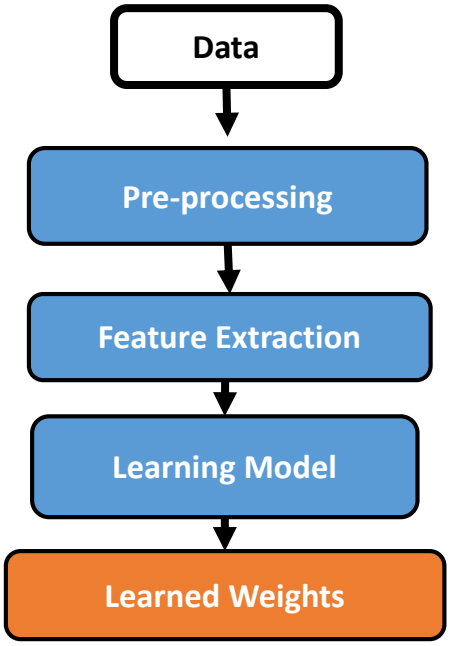
This paper developed various CNNs for a facial expression recognition problem and evaluated their performances using different post-processing and visualization techniques. The results demonstrated that deep CNNs are capable of learning facial characteristics and improving facial emotion detection. Also, the hybrid feature sets did not help in improving the model accuracy, which means that the convolutional networks can intrinsically learn the key facial features by using only raw pixel data.

8. Methodology

The whole task has been divided into 3 modules, each set performing a set of operations.

1. **Emotion Recognition using Speech Signals**
2. **Emotion Recognition using Facial Expressions**
3. **Fusion of the results.**

8.1 Emotion Recognition using Speech Signals: By analysing Table 1 [9], it seems that the classification is easy but unfortunately the problem is much complex. Looking closer at the studies of Murray and Arnott [9], one can notice that they are to some extent contradicting. The acoustic features of the acoustic effects of anger, joy and fear are very similar. For carrying out the required task we propose 3 different approaches. Figure shows the basic training architecture for our approaches.



8.1.1 Data Pre-Processing:

One of the major problem that occurs frequently in processing the audio clips is of the length. Unlike images which can be rescaled and cropped to a desired size, the audio clips vary in temporal extent and can't be easily processed on the fixed size architecture. In our case the audio clips vary between 2.0 and 3.0 seconds. All the audio clips are converted to 2.5 seconds by padding zeros on both ends or by clipping data from both ends resulting in 110250 (44100 * 2.5) sampling points for each time series audio clip. Since the analysis of signals in frequency domain is a bit easier and also it is easy to understand the behaviour and the pattern in the signal, the Spectrograms are computed for each of the clips using the python library librosa [15] with the window size of 512 samples, 50% overlap. This results in a matrix of shape (96, 431) for each audio clip which are used for classification purpose.

8.1.2 Model 1 (using MFCC):

This approach uses MFCC for extracting features from the input data. Mel Frequency Cepstral Coefficient (MFCC) is used to get short term features from the signal. Mel frequency scale is the most widely used feature of the speech, with a simple calculation, good ability of the distinction, anti-noise and other advantages. MFCC in the low frequency region has a good frequency resolution, and the robustness to noise is also very good, but the high frequency coefficient of accuracy is not satisfactory. In our research, we extract the first 12-order of the MFCC coefficients. The basic steps involved to obtain MFCC features [3][4].

i. Frame the signal into short frames.
ii. For each frame calculate the periodogram estimate of the power spectrum.
iii. Apply the mel filter bank to the power spectra, sum the energy in each filter.
iv. Take the logarithm of all filter bank energies.

v. Take the DCT of the log filter bank energies.
vi. Keep DCT coefficients 1-12, discard the rest.

We have used Artificial Neural Networks (ANN) as a training model for classifying the emotion from the speech data. The input audio data was framed into windows and MFCC features were extracted on each window resulting in a 12 x 1 vector for each window, these were then inputted to the ANN architecture. The system resulted with an accuracy of **50.12%** on the used test dataset.

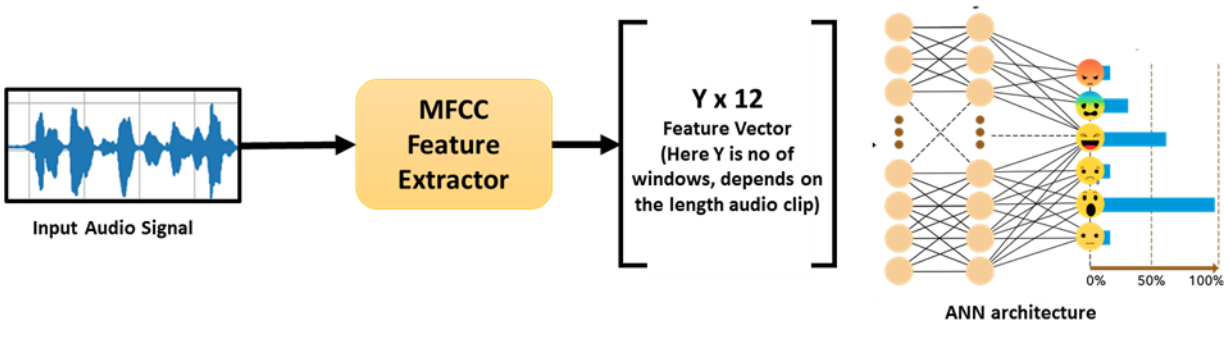


Fig. Approach 1 (MFCC + ANN)

8.1.3 Model 2 (using CNN):

Deep learning is a popular technique used in computer vision. We chose convolutional neural network (CNN) layers as building blocks to create our model architecture. CNNs are known to imitate how the human brain works when analysing visuals.

A typical architecture of a convolutional neural network will contain an input layer, some convolutional layers, some dense layers (aka. fully-connected layers), and an output layer. These are linearly stacked layers ordered in sequence. In Keras, the model is created as Sequential() and more layers are added to build architecture. The defined model is trained using the pre-processed training data with the batch size of 32 and 42 epochs. The model obtained an accuracy of **72.46%.** The architecture, confusion matrix and the results are displayed below.

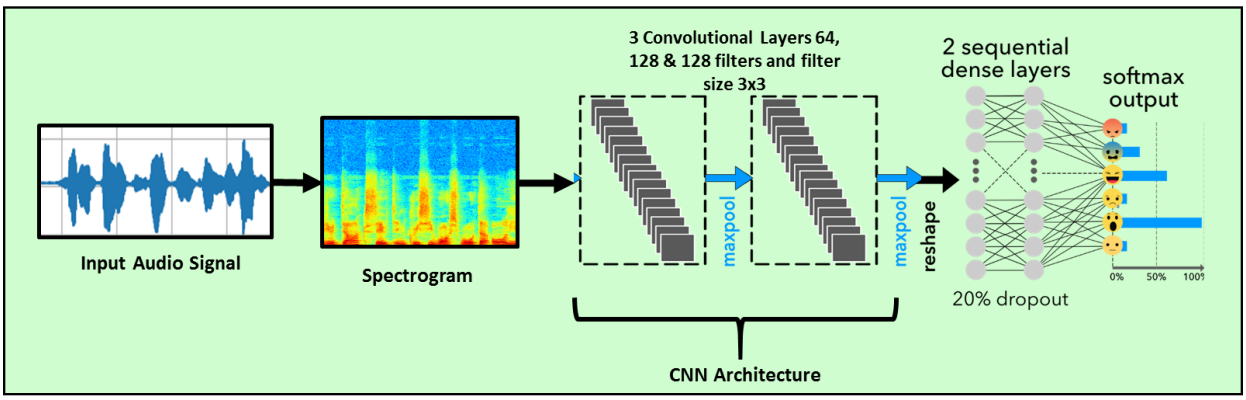


Fig. Approach 2 (Using CNN)

```
None
Layer (type)                 Output Shape              Param #   Connected to
=================================================================================================
convolution2d_1 (Convolution2D)  (None, 64, 98, 433)   640       convolution2d_input_1[0][0]
batchnormalization_1 (BatchNorma (None, 64, 98, 433)   256       convolution2d_1[0][0]
elu_1 (ELU)                  (None, 64, 98, 433)   0         batchnormalization_1[0][0]
maxpooling2d_1 (MaxPooling2D)  (None, 64, 49, 216)   0         elu_1[0][0]
dropout_1 (Dropout)          (None, 64, 49, 216)   0         maxpooling2d_1[0][0]
convolution2d_2 (Convolution2D)  (None, 128, 51, 218)  73856     dropout_1[0][0]
batchnormalization_2 (BatchNorma (None, 128, 51, 218)  512       convolution2d_2[0][0]
elu_2 (ELU)                  (None, 128, 51, 218)  0         batchnormalization_2[0][0]
maxpooling2d_2 (MaxPooling2D)  (None, 128, 17, 72)   0         elu_2[0][0]
dropout_2 (Dropout)          (None, 128, 17, 72)   0         maxpooling2d_2[0][0]
convolution2d_3 (Convolution2D)  (None, 128, 19, 74)   147584    dropout_2[0][0]
batchnormalization_3 (BatchNorma (None, 128, 19, 74)   512       convolution2d_3[0][0]
elu_3 (ELU)                  (None, 128, 19, 74)   0         batchnormalization_3[0][0]
maxpooling2d_3 (MaxPooling2D)  (None, 128, 4, 18)    0         elu_3[0][0]
dropout_3 (Dropout)          (None, 128, 4, 18)    0         maxpooling2d_3[0][0]
flatten_1 (Flatten)          (None, 9216)          0         dropout_3[0][0]
output1 (Dense)              (None, 2048)          18876416  flatten_1[0][0]
dropout_4 (Dropout)          (None, 2048)          0         output1[0][0]
output2 (Dense)              (None, 1024)          2098176   dropout_4[0][0]
dense_1 (Dense)              (None, 7)             7175      output2[0][0]
=================================================================================================
Total params: 21,205,127
Trainable params: 21,204,487
Non-trainable params: 640
```

```
Test score: 0.968174676273
Test accuracy: 0.724637680814
```

```
988/988 [==============================] - 30s - loss: 0.1435 - acc: 0.9484 - val_loss: 1.1864 - val_acc: 0.6928
Epoch 42/42
988/988 [==============================] - 29s - loss: 0.1029 - acc: 0.9706 - val_loss: 0.9682 - val_acc: 0.7246
/home/anirudh/.conda/envs/env_py2/lib/python2.7/site-packages/keras/models.py:697: UserWarning: The "show_accuracy"
l at compile time:
`model.compile(optimizer, loss, metrics=["accuracy"])`
  warnings.warn('The "show_accuracy" argument is deprecated, '
Test score: 0.968174676273
Test accuracy: 0.724637680814
```

	angry	disgust	fear	happy	neutral	sad	surprise
angry	49	0	0	7	1	0	1
disgust	4	33	4	5	4	3	0
fear	2	0	21	6	0	5	5
happy	10	4	3	33	2	0	2
neutral	0	1	0	0	52	6	0
sad	0	1	3	1	5	34	0
surprise	2	0	4	4	0	0	28

Confusion Matrix

8.1.4 Model 3 (using Spectrogram, CNN & LSTM):

Normal CNNs lack memory and do not consider the effects over the time. Because of which they lose the information about the past data. To overcome this we use LSTMs to use their memory capability.

The input signal is processed to fix the variable length problem and compute the spectrogram of the signal. The model first extracts some features using the 4 convolutional layers and the data is then reshaped to adjust the data with time. This data is further processed with the LSTM layers. The accuracy obtained using this model is **74.49%.**

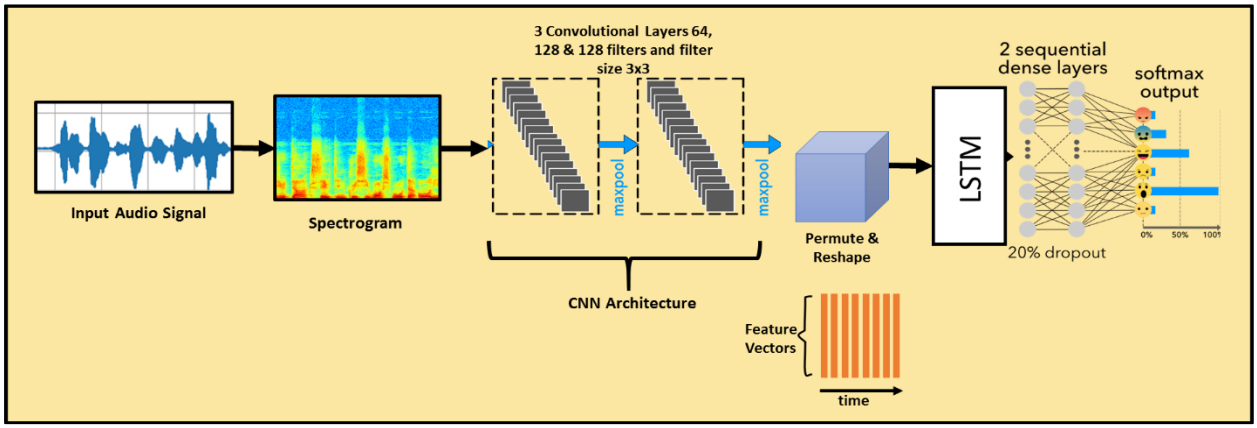


Fig. Approach 3 (Using CNN + LSTM)

elu_2 (ELU)	(None, 128, 51, 218)	0	batchnormalization_2[0][0]
maxpooling2d_2 (MaxPooling2D)	(None, 128, 17, 72)	0	elu_2[0][0]
dropout_2 (Dropout)	(None, 128, 17, 72)	0	maxpooling2d_2[0][0]
convolution2d_3 (Convolution2D)	(None, 128, 19, 74)	147584	dropout_2[0][0]
batchnormalization_3 (BatchNorma	(None, 128, 19, 74)	512	convolution2d_3[0][0]
elu_3 (ELU)	(None, 128, 19, 74)	0	batchnormalization_3[0][0]
maxpooling2d_3 (MaxPooling2D)	(None, 128, 4, 18)	0	elu_3[0][0]
dropout_3 (Dropout)	(None, 128, 4, 18)	0	maxpooling2d_3[0][0]
convolution2d_4 (Convolution2D)	(None, 128, 6, 20)	147584	dropout_3[0][0]
batchnormalization_4 (BatchNorma	(None, 128, 6, 20)	512	convolution2d_4[0][0]
elu_4 (ELU)	(None, 128, 6, 20)	0	batchnormalization_4[0][0]
maxpooling2d_4 (MaxPooling2D)	(None, 128, 1, 5)	0	elu_4[0][0]
dropout_4 (Dropout)	(None, 128, 1, 5)	0	maxpooling2d_4[0][0]
permute_1 (Permute)	(None, 5, 128, 1)	0	dropout_4[0][0]
reshape_1 (Reshape)	(None, 5, 128)	0	permute_1[0][0]
gru1 (LSTM)	(None, 5, 32)	20608	reshape_1[0][0]
gru2 (LSTM)	(None, 32)	8320	gru1[0][0]
output1 (Dense)	(None, 16)	528	gru2[0][0]
output2 (Dense)	(None, 16)	272	output1[0][0]
dense_1 (Dense)	(None, 7)	119	output2[0][0]

Total params: 401,303
Trainable params: 400,407
Non-trainable params: 896

```
Test score: 0.9643366737591
Test accuracy: 0.7449871345
```

	angry	disgust	fear	happy	neutral	sad	surprise
angry	49	0	0	7	1	0	1
disgust	2	35	4	4	5	3	0
fear	2	0	23	5	0	4	5
happy	8	3	3	36	2	0	2
neutral	0	1	0	0	54	4	0
sad	0	1	4	1	5	33	0
surprise	2	0	5	4	0	0	27

Confusion Matrix

8.2. Emotion Recognition using Facial Expression:

This module has already been developed in *the last semester project work*, with the accuracy of **61.27%.** The model was trained using the FER2013 Image dataset. The CNN model consisted of total 6 convolutional layers namely, 2 layers of 32 3x3 filter, 2 layers of 64 3x3 filters and 2 layers of 128 3x3 filters, with Maxpool of 2x2 & dropout of 10% after each layer followed by a dropout of 40% and 2 dense layers with 2048 neurons and 50% dropout. The total parameters (neurons) used are 7394247.

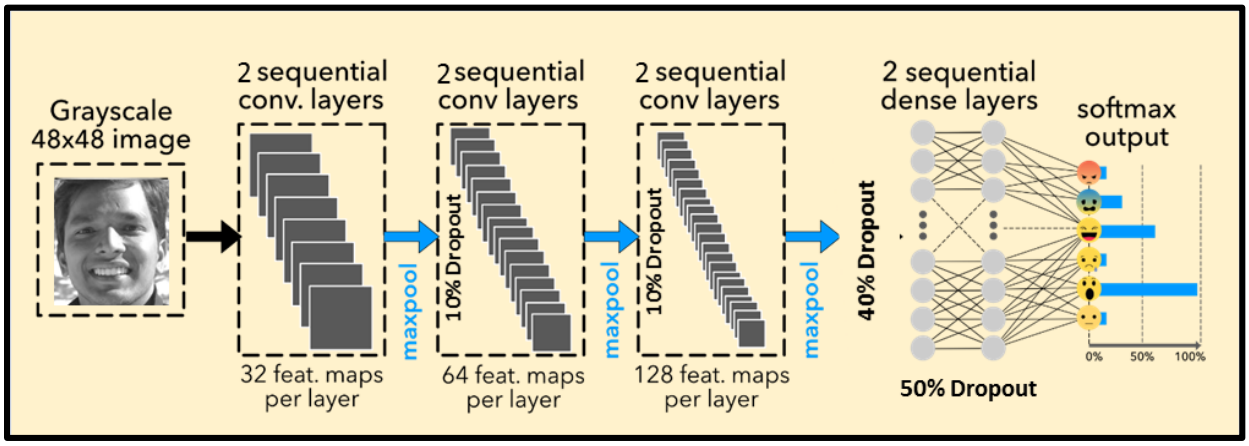


Fig. CNN architecture for Facial Expression Recognition

Layer (type)	Output Shape	Param #	Connected to
convolution2d_1 (Convolution2D)	(None, 3, 50, 32)	13856	convolution2d_input_1[0][0]
convolution2d_2 (Convolution2D)	(None, 5, 52, 32)	9248	convolution2d_1[0][0]
maxpooling2d_1 (MaxPooling2D)	(None, 2, 26, 32)	0	convolution2d_2[0][0]
dropout_1 (Dropout)	(None, 2, 26, 32)	0	maxpooling2d_1[0][0]
convolution2d_3 (Convolution2D)	(None, 4, 28, 64)	18496	dropout_1[0][0]
convolution2d_4 (Convolution2D)	(None, 6, 30, 64)	36928	convolution2d_3[0][0]
maxpooling2d_2 (MaxPooling2D)	(None, 3, 15, 64)	0	convolution2d_4[0][0]
dropout_2 (Dropout)	(None, 3, 15, 64)	0	maxpooling2d_2[0][0]
convolution2d_5 (Convolution2D)	(None, 5, 17, 128)	73856	dropout_2[0][0]
convolution2d_6 (Convolution2D)	(None, 7, 19, 128)	147584	convolution2d_5[0][0]
maxpooling2d_3 (MaxPooling2D)	(None, 3, 9, 128)	0	convolution2d_6[0][0]
dropout_3 (Dropout)	(None, 3, 9, 128)	0	maxpooling2d_3[0][0]
flatten_1 (Flatten)	(None, 3456)	0	dropout_3[0][0]
dense_1 (Dense)	(None, 2048)	7079936	flatten_1[0][0]
dropout_4 (Dropout)	(None, 2048)	0	dense_1[0][0]
dense_2 (Dense)	(None, 7)	14343	dropout_4[0][0]

Total params: 7394247

```
Test score: 1.08653423576
Test accuracy: 0.612705489002
```

	angry	disgust	fear	happy	sad	surprise	neutral
angry	239	3	41	44	70	12	58
disgust	14	23	6	4	7	0	2
fear	32	3	205	30	113	36	77
happy	22	1	13	744	30	17	68
sad	74	2	55	50	315	8	149
surprise	11	2	47	23	8	307	17
neutral	45	1	29	61	99	6	366

Confusion Matrix

8.3. Fusion of the results

Both the modules developed above are considered and their results are fused using the computed F-Scores. F-Scores are calculated using the confusion matrix of each modality.

According to definition, *The F score is also known by F-measure or F1-score. The test's accuracy can be measured in terms of F-score as it considers both the precision as well as the recall to compute the score of the test.*

- **Precision**: The ratio of number of true positive results and number of all the positive results is known as precision.

$$\text{Precision} = \frac{\mathbf{TP}}{\mathbf{TP + FP}}$$

TP=True Positive
FP=False Positive

- **Recall**: The ratio of number of true positive results and the number of true positive results and the false negative results calculated by the model is known as Recall.

$$\text{Recall} = \frac{\mathbf{TP}}{\mathbf{TP + FN}}$$

TP=True Positive
FN=False Negative

The F-score is further calculated using the harmonic mean of recall and precision. The value lies between 0 & 1, where 0 is considered worst and 1 is considered best.

$$\text{F-Score} = \frac{\mathbf{2 * (recall * precision)}}{\mathbf{recall + precision}}$$

The F-Scores calculated for both the modalities are specified in the table below:

F1 Scores		
Emotions	**Images**	**Speech**
Angry	0.5308	0.9034
Disgust	0.5748	0.4864
Fear	0.4581	0.5194
Happy	0.8113	0.6024
Neutral	0.5302	0.7880
Sad	0.4765	0.75
Surprise	0.7623	0.6236

Based on the computed F-Scores the Decision matrix is generated and the decision are made by overlooking the decision matrix.

Images	angry	disgust	fear	happy	neutral	sad	surprise
angry	angry	angry	angry	happy	neutral	sad	surprise
disgust	angry	disgust	disgust	happy	neutral	sad	surprise
fear	angry	disgust	fear	happy	neutral	sad	surprise
happy	angry	happy	happy	happy	happy	happy	happy
neutral	angry	neutral	neutral	happy	neutral	sad	surprise
sad	angry	disgust	fear	happy	neutral	sad	surprise
surprise	angry	surprise	surprise	surprise	neutral	sad	surprise

Fig. Decision Matrix

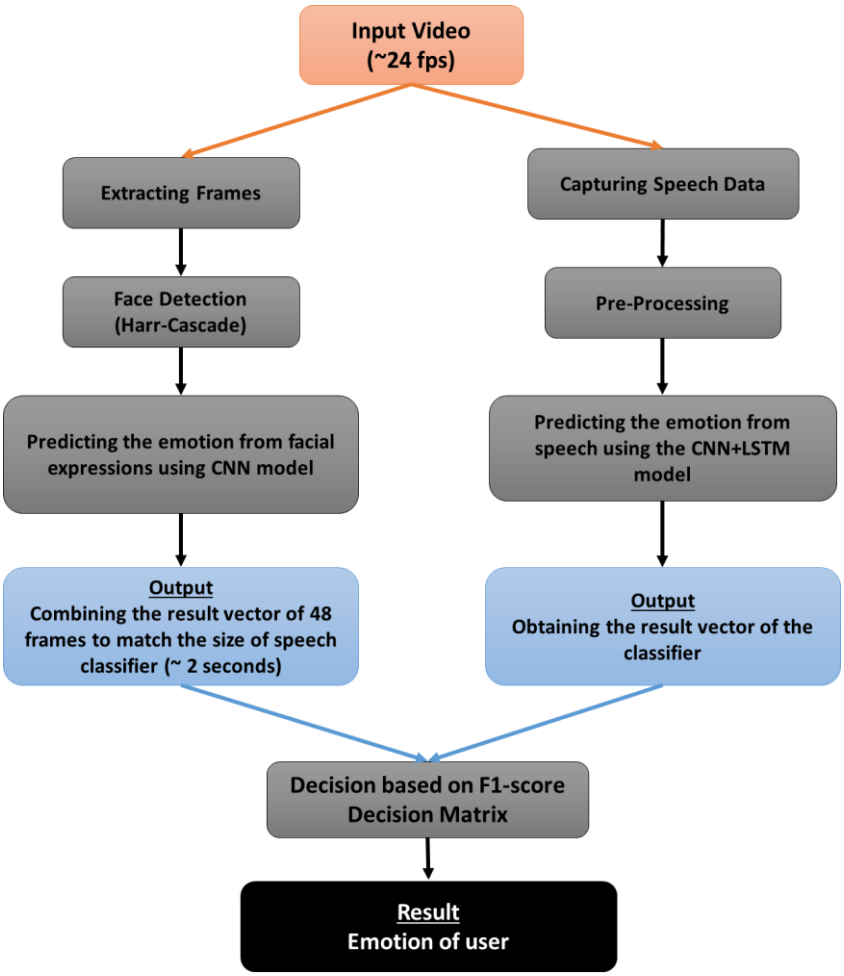
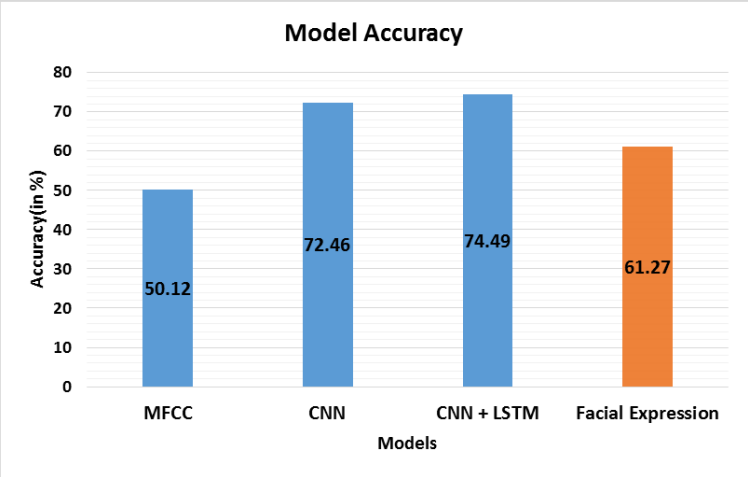


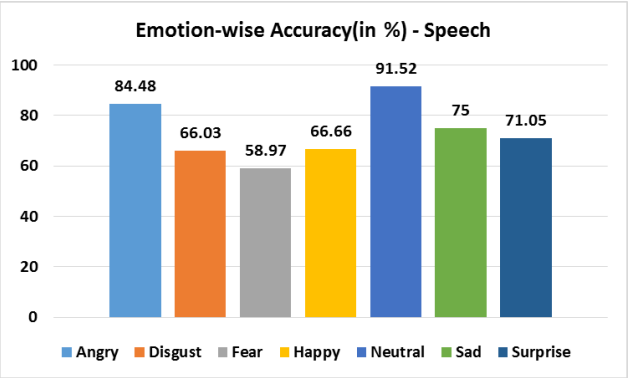
Fig: Flow Chart

9. Results and Analysis

As it turns out the final model (LSTM + CNN) has the **best validation accuracy of 74.49%** for speech signal classification. We further tried and tested the validity of the model by training it on different datasets. The results are provided in the table below.



Dataset	*Accuracy*
SAVEE	60.19%
IIIT A	80.93%
SAVEE + IIIT A	74.49%



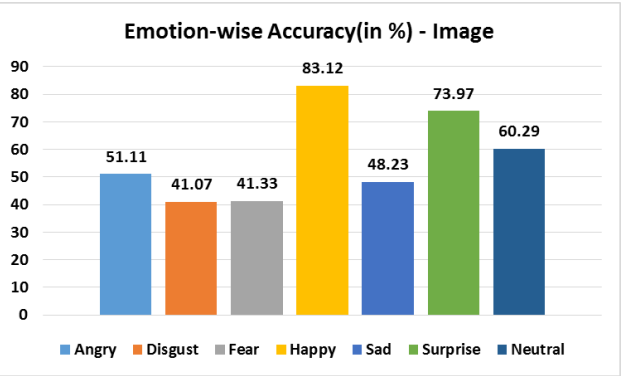


Fig: Accuracy Graphs

Applications Developed:

During the course of this project, 4 different python applications are developed to analyse the video and audio separately as well as together.

i. Emotion Classification using speech signal from audio file
ii. Emotion Classification using speech signal from live microphone audio
iii. Real Time Webcam Video Classification using Facial Expression
iv. Emotion Classification using fusion of speech and audio signals from video files

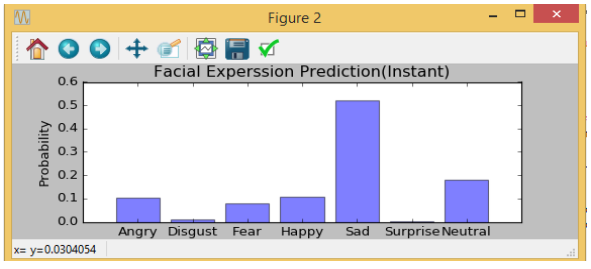
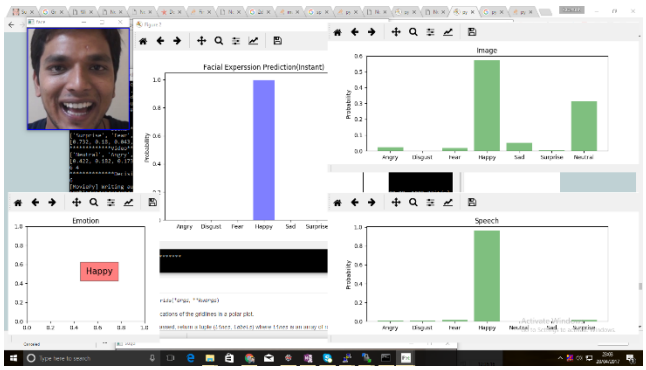


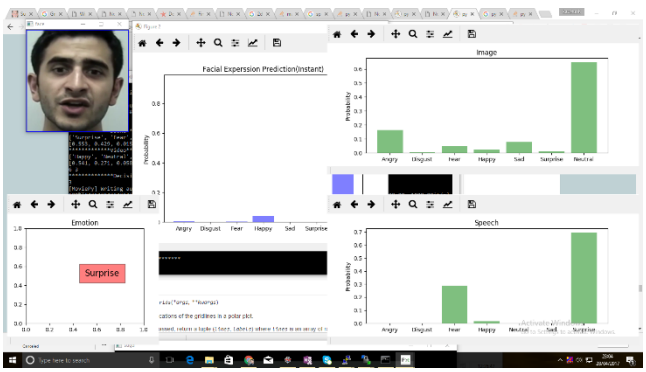
Fig: A bar graph showing the result probabilities of various emotions at an

```
t1.py | test_audio_file.py | test_file.py | untitled0.py | untitled3.py | audio_video_file.py

34
35 expressions = "Angry", "Disgust","Fear", "Happy","Sad", "Surprise", "Neutral"
36
37 def find_exp(score) :
38     yx = zip(score[0,:],expressions)
39     yx.sort(reverse=True)
40     y1 = np.empty([7], dtype='float32')
41     x1 = [x for y,x in yx]
42     y1 = [round(y,3) for y,x in yx]
43
44     return x1,y1
45
46
47
48
49 ################################
50 K.set_image_dim_ordering('th')
51
52 model= load_model('model_221.h5')
53 print ('Model Loaded....')
54
55 model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
56 model.load_weights('my_model_221.h5')
57 print ('Weights Loaded....')
58
59 ######################################
60
61
62
63 ##################VIDEO DATA
64
65
66 face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
67 cap = cv2.VideoCapture('MAH01715.MP4')
68 ans = []
69 time = []
70 p = 0
71 i = 0
72 c = 0
73 t = 0
74 x_test = np.empty([1,1,48,48])
75 f = open('emotions.txt','w')
76 f.close()
77 while(cap.isOpened()):
78
79     ret, frame = cap.read()
80     #gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
```

```
Editor - C:\Users\Incarnation\Pictures\Screenshots\CNNandRNN_saving_Weights.py

test_file.py | untitled0.py | untitled3.py | audio_video_file.py | test_video_file.py | CNNandRNN_saving_We

58 #plt.imshow(X_train[5][0], interpolation='nearest')
59 #plt.show()
60
61
62 print("--------------test data----------")
63 print (X_test.shape)
64 print (Y_test.shape)
65 print (test_size)
66 print (test_rows)
67 print (test_cols)
68
69
70 X_train = X_train.astype("float32")
71 X_test = X_test.astype("float32")
72
73 Y_train = np_utils.to_categorical(Y_train,7)
74 num_classes = Y_train.shape[1]
75 Y_test = np_utils.to_categorical(Y_test,7)
76
77
78
79 print("############model###############")
80 channel_axis = 1
81 freq_axis = 2
82 time_axis = 3
83 np.random.seed(1337)  # for reproducibility
84 K.set_image_dim_ordering('th')
85
86
87 model = Sequential()
88
89 model.add(Convolution2D(64, 3, 3, border_mode='full', input_shape=(1, train_rows, train_cols)))
90 model.add(BatchNormalization(axis=channel_axis, mode=0))
91 model.add(ELU())
92 model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))
93 model.add(Dropout(0.1))
94
95 model.add(Convolution2D(128, 3, 3, border_mode='full'))
96 model.add(BatchNormalization(axis=channel_axis, mode=0))
97 model.add(ELU())
98 model.add(MaxPooling2D(pool_size=(3, 3), strides=(3, 3)))
99 model.add(Dropout(0.1))
100
101 model.add(Convolution2D(128, 3, 3, border_mode='full'))
102 model.add(BatchNormalization(axis=channel_axis, mode=0))
103 model.add(ELU())
```

Fig: Code Snippets





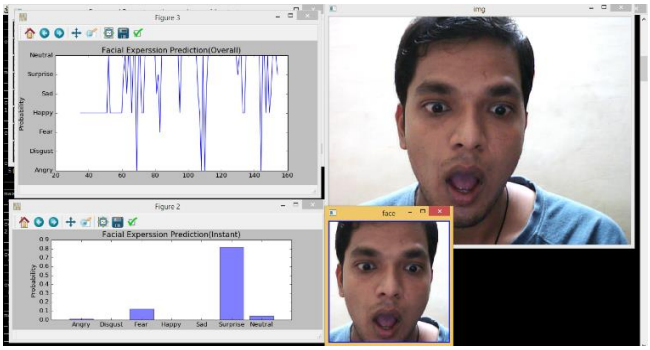


Fig: Demo of various developed applications

10. Technical Requirements

- Anaconda
- Nvidia GPU
- Keras
- PyAudioAnalysis
- Librosa
- PyCharm
- Theano

11. Conclusion

We developed various models for speech emotion detection and evaluated their performances using different post-processing and visualization techniques. The results obtained from speech emotion detection demonstrated that deep CNNs applied on spectrograph, are capable of learning audio features and improve speech emotion recognition process. Also, the hand tailored MFCC features did not help much in improving the accuracy. It was further observed that CNNs when combined with LSTM further improved the accuracy of the model. We then combined the speech emotion detection model with facial expressions recognition model developed in the last semester, and created a combined model that takes audio-video clip as input and provides an emotion as the output in real time.

12. References

[1] http://kahlan.eps.surrey.ac.uk/savee/Database.html

[2] https://docs.python.org/3/

[3] Zhang Wanli and Li Guoxin "The Research of Feature Extraction Based on MFCC for Speaker Recognition "

[4] http://practicalcryptography.com/miscellaneous/machine-learning/guide-mel-frequency-cepstral-coefficients-mfccs/

[5] LeCun, Yann, and Yoshua Bengio. "Convolutional networks for images, speech, and time series." *The handbook of brain theory and neural networks*3361.10 (1995): 1995.

[6] http://cs231n.github.io/convolutional-networks/

[7] Mollahosseini, Ali, David Chan, and Mohammad H. Mahoor. "Going deeper in facial expression recognition using deep neural networks." *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2016.

[8] Hasim Sak, Andrew Senior, Franc¸oise Beaufays Google, USA "Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling"

[9] I. R. Murray and J. L. Arnott, "Toward the simulation of emotion in synthetic speech: A review of the literature on human vocal emotion," Journal of the Acoustical Society of America, vol. 93, no. 2, pp. 1097–1108, February 1993

[10] Namrata Anand, Prateek Verma "Convoluted Feelings:Convolutional and recurrent nets for

detecting emotion from audio data”
[11] H. Lee, P. Pham, Y. Largman, A. Ng, A. Culotta, "Unsupervised feature learning for audio classification using convolutional deep belief networks" in Advances in Neural Information Processing Systems 22, MIT Press, pp. 1096-1104, 2009.
[12] Alizadeh, Shima, and Azar Fazel. "Convolutional Neural Networks for Facial Expression Recognition."
[13] http://colah.github.io/posts/2015-08-Understanding-LSTMs/
[14] S. Haq, P. Jackson, and J. Edge, “Speaker-dependent audio-visual emotion recognition,” in AVSP, Norwich, UK, Sept. 2009, pp. 53–58
[15] https://github.com/librosa/librosa

13. Panel Comments & Suggestions