**Table of Contents**

**Chapter 1: Introduction**

**1.1 Introduction  
 1.2 Motivation  
 1.3 Aim & Objectives  
 1.4 Project Scope  
 1.5 Methodology & Project Plan**

**Chapter 2: Related Work**

**2.1 Introduction  
 2.2 Background  
 2.3 Feature Extraction & Data Preprocessing  
 2.5 Audio & Video Emotion Recognition Datasets**

**Chapter 3: Audio Emotion Recognition system**

**3.1 Introduction  
 3.2 Feature Extraction  
 3.2.1 MFCC & Spectrogram  
 3.3 Model Work  
    3.3.1 Two Parallel CNN Model  
     3.3.2 CNN Model  
    3.3.3 Machine Learning Models**

**Chapter 4: Video Emotion Recognition system**

**4.1 Introduction  
 4.2 Feature Extraction & Preprocessing  
 4.3 Model Work  
    4.3.1 LRCN Model  
  4.3.2 ConvLSTM Model  
 4.5 Conclusion**

**Chapter 5: Conclusion & Future Work**

**5.1 Conclusion  
 5.2 Future Work**

**Abbreviations**

**Chapter 1 Introduction**

### 1.1 Introduction

Emotions are difficult to recognize because they depend on context, facial expressions, speech and gestures. The latest technologies based on data-centric techniques have come a long way, but modern day machine learning algorithms is still a very difficult task despite recent efforts to augment artificial intelligence techniques to consider context. For this reason, people who are working with emotion recognition tend to work with individual aspects such as identifying a person’s emotions based purely on looking at their faces, through speech, or through the vocabulary used. “If there is so much room for error when it comes to reading a person’s facial expressions, we must question how a machine can ever be programmed to get it right.”

Today, Emotion Recognition Systems can help firms vet job applicants, analyze the emotional impact advertisements have on people as well as helping police detect criminal suspects

Many systems are based on the work of the psychologist Paul Ekman (Ekman, 2011) who proposed seven basic emotions which he then changed to six that are recognized around the world: happiness, sadness, fear, anger, surprise and disgust. His research took him to the conclusion that the recognition of emotions was similar across cultures, and thus universal.

The purpose of this project is to try to recognize emotions through the combination of two techniques based on Paul Elkman’s study of the six basic emotions. Since the correlation between facial expression and emotion has been questioned due to the fact that different cultures have varying expectations for when it is appropriate to use a certain expression and in what manner, it was decided to add emotion detection through audio in order to have two different angles.

Taking all the above into account, the project will plan out in the following way: First, an emotion detection model through audio features will be trained by trying different combinations of datasets and neural networks to try and achieve the highest possible accuracy. Then the same will be done with video features (facial expressions). Once both models have been trained separately, they will be combined together into a new model in order to see whether a greater accuracy percentage is achieved together rather than separately.

### 1.2 Motivation

The main motivation to carry out this project is to learn new programming techniques as well as expanding new horizons in a field we have become increasingly fascinated with machine learning. What we wish to obtain when developing this project is a good accuracy in emotion detection through video and audio separately and a better result when combining the two models together. In theory a model that recognizes emotions through video and audio at the same time should learn better and therefore be more accurate than two individual models.

### 1.3 Aim

The main aim of our project is to Help computers differentiate between six different emotions using audio-visual data.

### 1.4 Objectives

* Focusing on related work in the area of video-audio emotion recognition.
* Determining suitable feature extraction methods for audio files.
* Apply Facial landmark recognition for video.
* Investigating the most suitable classification model for audio and video emotion recognition.
* Perform fusion between two classification models (audio & video).

### 1.5 Project Scope

Machine learning /deep learning system for emotion recognition from video and sound on video-audio dataset to help in detecting human emotion by video and sound by high efficiency.

### 1.6 Methodology

* **Planning and Literature review:** Reading papers related to emotion recognition from video, sound and fusion multimodal to build background about this field
* **Explore datasets:** Searching for video-sound datasets or implementing our own dataset.
* **Structuring Audio Models, techniques and testing:** know more about most used models, algorithms and techniques for audio thenComparing between available techniques and methods to choose the most appropriate among them then implementing audio model and testing
* **Structuring Video Models, techniques and testing:** know more about most used models, algorithms and techniques for video thenComparing between available techniques and methods to choose the most appropriate among them then implementing video model and testing
* **Testing system performance:** Testing different methods and making comparisons between the results.

## Chapter 2 (Related Work)

### 2.1 Introduction

We go over a background and some principles in Performance Metrics and classification and some related work studies. Emotion recognition can be used in different applications like companies that want to improve customer experience, helping blind people to read facial expressions and helping robots interact more intelligently with people and analyze their emotions.

### 2.2 Background

Emotion recognition is an important research field for Human-Computer Interaction. Emotion recognition has many applications. For example, instead of filling out a lengthy survey about how you feel at each point watching an educational video or advertisement, you can consent to have a camera watch your face and listen to what you say. Other uses include helping children with autism, helping people who are blind to read facial expressions, helping robots interact more intelligently with people, and monitoring signs of attention while driving in an effort to enhance driver safety. Audio-Video Emotion Recognition is now attacked with Deep Neural Network modeling tools. Some papers introduce cases of the superiority in multi-modality over audio-only or video-only modality and other studies introduce models of fusion both audio and video.

### 2.3 Feature Extraction and Data Preprocessing

We will explain how the audio files have been processed as well as the video files and the procedure that has been followed to extract the desired features.

#### 2.3.1 Data Preprocessing

Data preprocessing is a crucial step that helps enhance the quality of data to promote the extraction of meaningful insights from the data. Data preprocessing refers to the technique of preparing (cleaning and organizing) the raw data to make it suitable for a building and training Machine Learning models [1] [2].

Audio signal has to be preprocessed before the features extraction step which will be mentioned in this section. Preprocessing includes making segmentations of audio files, resampling their sample rate or mixing it with other datasets. Making segmentations of an audio file means dividing it into chunks to have more data to train on or to know the effect of signal time on results. [1]

Video preprocessing techniques are required to reduce the complexity of the succeeding steps consisting of video text detection, localization, segmentation, recognition, and script identification. [2]

#### 2.3.2 Feature Extraction

Audio feature extraction is a necessary step in audio signal processing. It removes unwanted noise and balances the time-frequency ranges by converting digital and analog signals. to describe audio, extract more than one feature. There are several ways of extracting features from an audio file such as Mel-frequency cepstral coefficients (MFCC) and Mel spectrograms (Mel Spectrogram) are commonly used in speech recognition and speaker recognition systems. [1] Video Feature extraction means to find out the “point of interest” or differentiating frames of video. These features are consistent over several video frames of the same scene and after the video scene is changed the value of these features are changed. all the frames from each video are processed independently. These undergo two main steps, the first one consists of detecting the facial landmarks of the face and the second one consists of aligning the face. Therefore we can use features to classify videos[3].

**MFCC**

The set of Mel-Frequency Cepstral Coefficients is a cepstral representation of the audio signal obtained based on the Mel-scaled spectrum [4]. The most used MFCCs consists of between 13 to 45 coefficients.

**Mel Spectrogram**

A spectrogram is an image representation of the waveform signal, it shows its frequency intensity range over time, it can be very useful when evaluating the signal’s frequency distribution over time. Mel-Spectrogram allows plotting amplitude on frequency vs time graph on a Mel scale [5]. Spectrogram are commonly used to display frequencies of sound waves produced by humans, animals, etc as recorded by microphones. Spectrograms are used extensively in the fields of [sonar,](https://en.wikipedia.org/wiki/Sonar) [radar,](https://en.wikipedia.org/wiki/Radar) [speech processing](https://en.wikipedia.org/wiki/Speech_processing) and can be used to identify spoken words [phonetically.](https://en.wikipedia.org/wiki/Phonetics)

A spectrogram can be generated by an [optical spectrometer,](https://en.wikipedia.org/wiki/Optical_spectrometer) a bank of [band-pass filters,](https://en.wikipedia.org/wiki/Band-pass_filter) by [Fourier transform](https://en.wikipedia.org/wiki/Fourier_transform) or by a [wavelet transform](https://en.wikipedia.org/wiki/Wavelet_transform) in which case it is also known as a scalogram. A spectrogram is usually depicted as a [heat map,](https://en.wikipedia.org/wiki/Heat_map) i.e., as an image with the intensity shown by varying the [color](https://en.wikipedia.org/wiki/Colour) or [brightness](https://en.wikipedia.org/wiki/Brightness) [6].

**Facial Landmarks**

Used to localize and represent specific parts of the face by using shape prediction methods. Detecting facial landmarks can be divided into two steps: Localizing the face in the image and detecting the key facial landmarks on the region of interest. OpenCV is used to detect facial landmarks in an image [3]

**Aligning The Face**

Face alignment consists in identifying the geometric structure of faces in digital images and obtaining a canonical alignment of the face based on translation, scale, and rotation. [3]. [7]

### 2.4 Performance Metrics

Once the model has been trained, the accuracy and loss graphs are plotted.

The loss learning curve measures our model error, or “how bad our model is doing”, therefore, the lower the loss becomes, the better the model’s performance will be. The accuracy learning curve captures the model’s performance, so the higher it is, the better the model becomes.

#### 2.4.1 Accuracy

It is the simplest metric to use and implement and is defined as the number of correct predictions divided by the total number of predictions, multiplied by 100. Can be implemented by comparing ground truth and predicted values in a loop or simply by utilizing the scikit-learn module to do the heavy lifting.

For example, Start by just importing the *accuracy\_score* function from the *metrics* class. Then, just by passing the ground truth and predicted values we can determine the accuracy of our model [7].

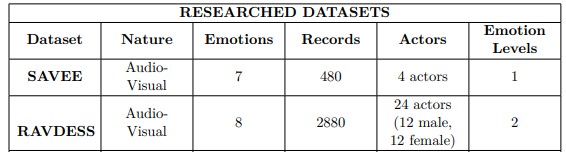
#### 2.4.2 Confusion Matrix

It is a tabular visualization of the ground-truth labels versus model predictions. Each row of the confusion matrix represents the instances in a predicted class and each column represents the instances in an actual class[7].

### 2.5 Audio Visual Emotion Recognition Dataset

An early step of the project was to carry out a thorough research of audiovisual datasets and gather enough information on them for afterwards being able to select those most suited for the problems to solve. Two of the most used datasets were chosen.

*Table 2-1 : Show Datasets details*



#### 2.5.1 RAVDESS

The Ryerson Audio-Visual Database of Emotional Speech and Songs a validated multimodal database of emotional speech and song. The database is gender balanced consisting of 24 professional actors (12 male,12 female), vocalize two statements in a North American accent. Speech includes calm, happy, sad, angry, fearful, surprise, and disgust expressions, and song contains calm, happy, sad, angry, and fearful emotions Each one of these expressions is produced at two levels of emotional intensity (normal and strong), with an additional neutral expression and can be obtained in three different formats: Audio-only, AudioVideo and Video-only. This dataset was downloaded in both Audio only and Audio-Video formats; one for emotion recognition through audio and the other for emotion recognition through video [8].

The speech dataset used contains a total of 1440 files (60 trials x 24 actors) for the eight emotions mentioned above. While the song dataset contains a total of 1056 files (44 trials x 24 actors) for the five emotions. Each of the files has a unique filename consisting of a 7-part numerical identifier which define the different characteristics:



*Figure 2-1 : RAVDESS actor examples (source: (LIVINGSTONE &*

*RUSSO ,2018))*

* Modality (01 = full-AV, 02 = video-only, 03 = audio-only).
* Vocal channel (01 = speech, 02 = song).
* Emotion (01 = neutral, 02 = calm, 03 = happy, 04 = sad, 05 = angry, 06 = fearful, 07 = disgust, 08 = surprised).
* Emotional intensity (01 = normal, 02 = strong). NOTE: There is no strong intensity for the ’neutral’ emotion.
* Statement (01 = ”Kids are talking by the door”, 02 = ”Dogs are sitting by the door”).
* Repetition (01 = 1st repetition, 02 = 2nd repetition).
* Actor (01 to 24. Odd numbered actors are male, even numbered actors are female).

#### 2.5.2 SAVEE

The Surrey Audio-Visual Expressed Emotion (SAVEE) Database, was recorded for the purpose of developing automatic emotion recognition systems. It consists of recordings from only male actors (four actors) aged from 27 to 31 for seven emotions: Anger, disgust, fear, happiness, sadness, surprise and neutral. For each emotion, a total of 15 sentences were recorded; 3 common, 2 emotionspecific and 10 generic. For the neutral emotion, the three common and 12 emotion-specific sentences were also recorded adding up to 30 neutral sentences. In total, there are 480 utterances per speaker [9]



*Figure 2-2 : SAVEE actor frames examples (source (Jackson & Haq ,2015))*

It is important to mention that the recordings took place in a 3D vision laboratory during different periods of the year. The recordings made sure to avoid bias due to fatigue by dividing the text prompts into groups such that each group had sentences for each emotion. At the same time, a 3dMD dynamic face capture system was used to capture the 2D frontal color video and Beyerdynamic microphone signals. The main advantages for using this dataset since the beginning is the audio-visual nature together with the fact that it is the smallest dataset which makes it ideal to start with. It speeds up the process when trying new solutions and it is a good starting point for initial results.

Chapter 3 AUDIO Emotion Recognition System

### 3.1 Introduction

In this chapter, we will review the work we have done in the audio model, we will review our work in the data extraction process and how to choose the appropriate method for that project, and then we will review our work in building the model and how to modify it to reach a satisfactory result and high accuracy.

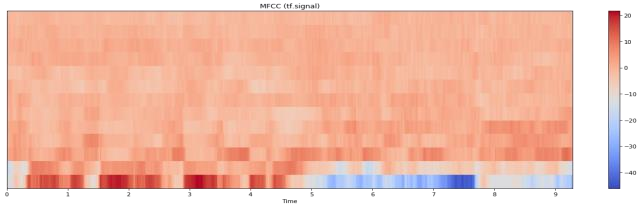
### 3.2 Preprocessing and feature extraction

The first step in any automatic sound recognition system is to extract features i.e., identify the components of the audio signal that are good for identifying the linguistic content and discarding all the other stuff which carries information like background noise. This phase is done after preprocessing which is applied to make us test our system on in different conditions, we obtain by manipulating our dataset.

In order to be able to choose the appropriate method for extracting Feature, we had to try the most famous methods available, so we chose two methods to prefer between them, MFCC and SPECTROGRAM, and this is due to what we found in the previous work (as we mentioned in the chapter two).

### 3.2.1 MFCC

Mel Frequency Cepstral Coefficients (MFCCs) are a feature widely used in automatic sound recognition. They were introduced by Davis and Mermelstein in the 1980's and have been state-of-the-art ever since. We transform the sound waveform into MFCCs representation as shown in (figure 3.1). The reason we chose this form is that it is the most usable and efficient one to the patterns of human recognition and preservation. The most used MFCCs consist of between 13 to 45 coefficients. The first 13 represent the basic feature of the wave, and the rest of the feature are redundant (even if you feed it into your model, you will get slight performance) as the lower order coefficients contain most of the information about overall spectral shape of the source. The zero order coefficient indicates the average power of input signal and the first-order coefficient represent the increasing levels of spectral details [1][4].

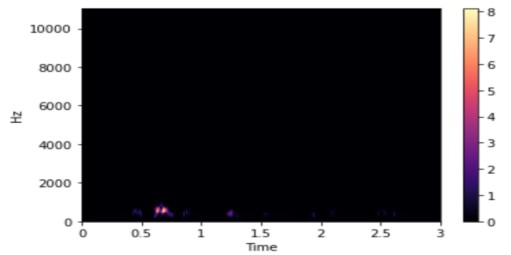


*Figure 3-1 :MFCC Coefficient*

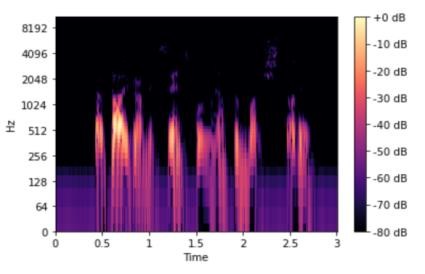
### 3.2.2 SPECTROGRAM

A spectrogram is an image representation of the waveform signal, it shows its frequency intensity range over time, it can be very useful when evaluating the signal’s frequency distribution over time [5].

The first thing that is done is to try and extract useful information from the digital representation of an audio signal. The Fourier transform is a mathematical formula that decomposes a signal into its individual frequencies and the frequency’s amplitude. In other words, it converts the signal from the time domain into the frequency domain, also called a spectrum. This is possible since any signal can be decomposed into a set of sine and cosine waves that add up to the original signal. To compute a power spectrogram, the function librosa.stft(y) is used (librosa development team, 2021), where ‘y’ is the time series obtained from the audio file. The Short-time Fourier Transform (STFT) represents a signal in the timefrequency domain by computing Discrete Fourier Transforms (DFT) over short overlapping windows. Therefore the function returns an array of Fourier transform coefficients (complex numbers). These coefficients give the phase and amplitude of the audio signal [6]. Since humans don’t perceive the phase very well, the signal is reduced to the amplitude and therefore the absolute number of the complex value is obtained as shown in (figure 3.5).



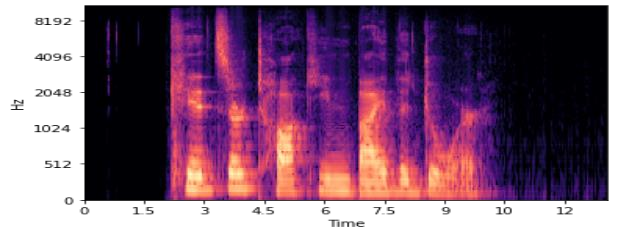
*Figure 3-5 : Show signal is reduced to the amplitude and therefore the absolute number of the complex value is obtained as shown here*



*Figure 3-6*

Not much can be seen and this is because most sounds humans hear are concentrated in very small frequency and amplitude ranges. Therefore, in order to observe something a small adjustment is made, the y-axis (frequency) is transformed to log-scale and the color-axis (amplitude) is transformed to decibels (librosa development teams, 2021). as shown in (figure 3.6).

Given the spectrogram obtained earlier, it is mapped on to the mel scale acquiring the mel-spectrogram (librosa’s development team, 2021). Both the y-axis (frequency), and the power (amplitude squared) axis are once again transformed to log scale and decibels respectively as shown in (figure 3.7).



*Figure 3-7*

The next section discusses our experiments trying different models with spectrogram feature extraction, the focus is not on the model itself, but on the spectrogram and the results it achieves in emotion recognition.

##### 3.3 Model Work

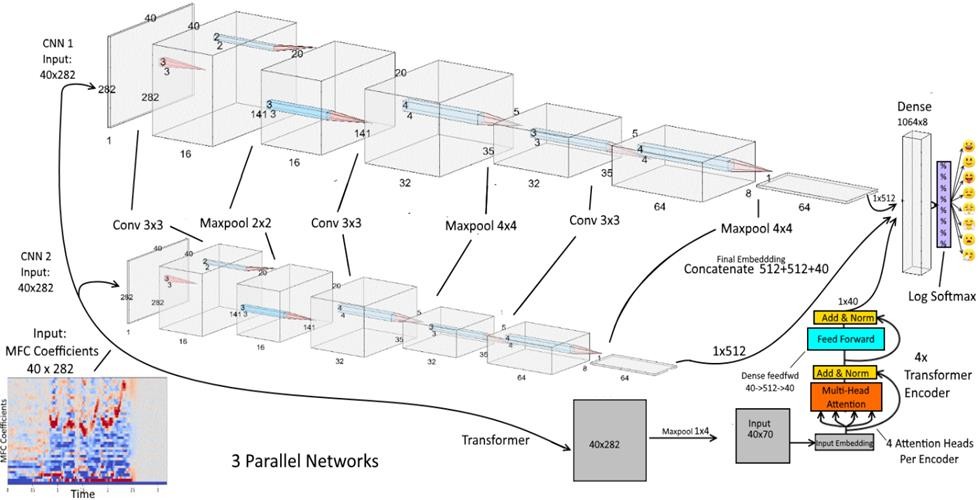
In order to build a model with good accuracy, we had to choose the appropriate model through previous work, so we chose two models that have the highest accuracy using the MFCC, Then we will make adjustments and improvements in order to achieve a satisfactory result and high accuracy

#### 3.3.1 Two Parallel CNN Model

We trained this model before in section 3.2.1.1 to choose the feature extraction method, now we will go in depth in this model in sections 3.3.1.1 and 3.3.1.2, and try to make changes to achieve higher accuracy in section 3.3.1.3.

##### 3.3.1.1 Model Architecture

The work in this reference [28] proposes two parallel convolutional neural networks (CNN) in parallel with a transformer encoder network to classify audio data. Model is shown in the figure below Figure 3.21.



*Figure 3-21 : Two parallel CNN model architecture*

We use 3x3 kernels in all 3 layers in both CNN blocks. The first layer only has a single input channel creating a 1x3x3 filter, with 16 output channels necessitating 16 such unique 1x3x3 filters with 1x3x3=9 weights per filter. The next layer has 16 input channels and 32 output channels, producing 32 unique 16x3x3 filters, each filter having 16x3x3 = 144 weights. That is, the second layer is applying 32 differently weighted 16x3x3 filters to an input volume of 16x20x141 (the 2x2 max pooled output of the first layer), producing an output feature map of

32x5x35 after 4x4 stride 4 max pooling. The last layer has 32 input channels, so a 32x3x3 filter, and 64 output channels, so 64 unique such filters with 32x3x3=288 weights each. The last layer produces an output feature map of 64x1x8 after 4x4 stride 4 max pooling.

**3.3.1.2 Model implementation**

After we know model Architecture in the previous section, in this section will know how the model works and its results.

We combine the CNN for spatial feature representation and the Transformer for temporal feature representation. Because of the sequential nature of the data, we will also use the Transformer to try and model as accurately as possible the temporal relationships between pitch transitions in emotions. CNNs with 2D convolutional layers are the gold standard for image processing, other than the recent advances in the Transformer for images. 2D convolution layers accept input feature maps in a (N,C,H,W) (batch size, channel, height, width) format. We have 7356 MFCC plots - 2452 native (RAVDESS speech & song) and 4904 noise augmented by (AWGN) augmentation - each MFCC plot is of shape 40x282 with 40 MFC coefficients representing different Mel pitch ranges, with 282 timesteps for each MFC coefficient. We can imagine MFCC plots to be a black and white image with 1 signal intensity channel. Our MFCC input feature tensor will thus be of shape (7356, 1, 40, 282) before splitting for training.trained on Ravadess speech and song data set , split data set into 80 % for train , 10 % for test and 10 % for validation, using **CrossEntropyLoss** as loss function , **SGD**as optimizer and learning rate **LR = 0.01** . Model achieved an average accuracy 73.23 ± 2 % test accuracy on 8 emotions using RAVDESS speech & song.

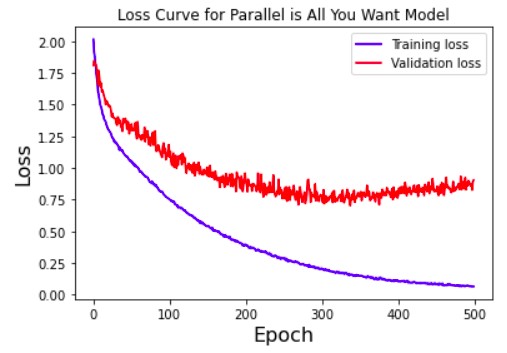
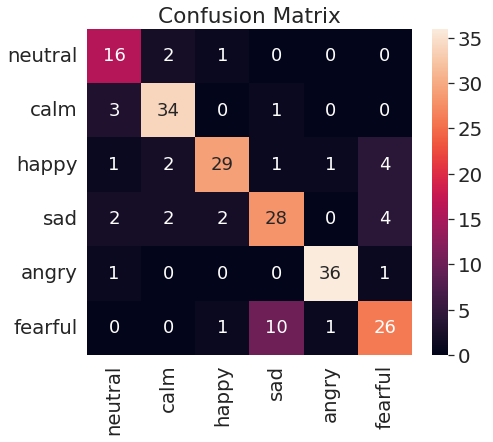
##### 3.3.1.3 Model Schemas

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **6** | **LR DECREASE WITH TIME**  **&**  **AUGMENTATION**  **On Train Only** | **RAVDESS**  **Speech**  **&**  **Song** | **6** | **80.86 %** |

**Schema 6:**

Change Learning Rate Dynamically at the middle of training from lr =

0.01 to lr = 0.001. This schema achieves accuracy 80.86 % on average.

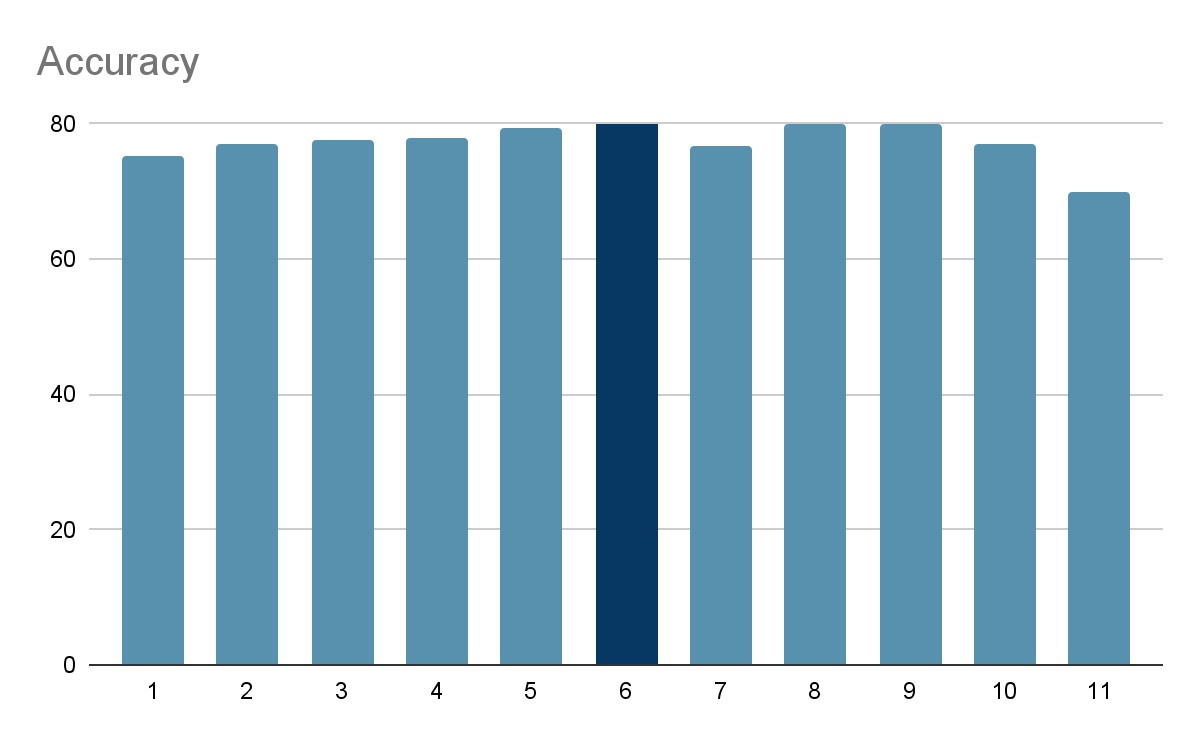


*Figure 3-28 : Schema 6 loss curve Figure 3-27 : Schema 6 Heatmap of confusion matrix*

##### 3.3.1.4 Conclusion

Best accuracy achieved on the two parallel CNN model on 6 emotions is

80.86% by Schema 6 .



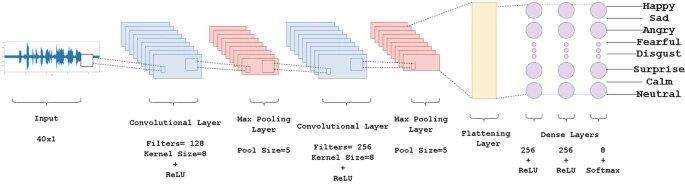
*Figure 3-40 : Chart shows accuracy achieved by Two parallel CNN model schemas*

#### 3.3.2 CNN Model

We trained this model before in section 3.2.1.1 to choose the feature extraction method, Now we will go in depth in this model in sections 3.3.2.1 and 3.3.2.2, and try to make changes to achieve higher accuracy in section 3.3.2.3.

##### 3.3.2.1 Model Architecture

The convolutional neural network model presented in this reference [10] consists of two convolution layers with kernel size of 8 and employing ReLU activation function. The first convolutional layer contains 128 filters, whereas the second contains 256 filters. Each of the convolutional layers is followed by a maxpooling layer with a pool size of 5. Followed by these layers are the flattening layer and three dense layers with the first two dense layers having 256 output perceptrons and employing ReLU activation function, whereas the third layer has eight output perceptrons employing softmax function. The following figure Fig 3.41 shows the architecture of the CNN model.



*Figure 3-41: CNN model architecture*

##### 3.3.2.2 Model implementation

In order to implement this model, We used Model architecture fig 3.41 and its hyper parameter table 3.3 mentioned in the reference Results achieved in the reference is 70.83% on 8 emotions. But The reference did not mention the optimizer used in this model so we are going to try optimizers from related work [chapter two] to choose suitable optimizer, The optimizers that will try are ADAM and RMS. but on 6 emotions on RAVDESS Speech & Song and no augmentation happened on the dataset

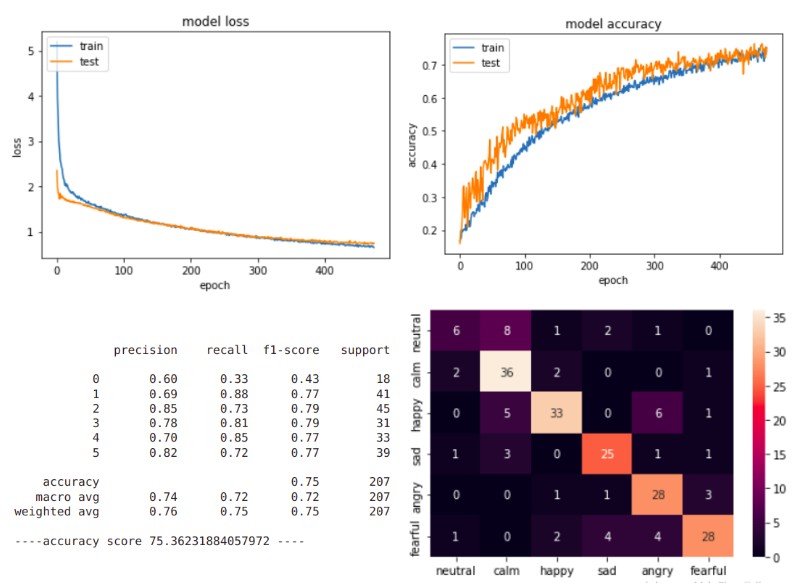
*Table 3-3 : Shows Hyper-parameters in the reference*

|  |  |
| --- | --- |
| **Hyper-parameter** | **Value** |
| Loss Function | Sparse Categorical cross entropy |
| Learning rate | 0.00002 |
| Dropout factor | 0.3 |
| Decay | 0.9 |

In order to implement this model. we used an early stopping function to override overfitting. Any experiment or modification is tried at least 4 times to calculate the average test accuracy. So in order to change the test and train set each time we run the experiment we shuffled the data. So The shuffle parameter is needed to prevent non-random assignment to train and test set. With shuffle equal True we split the data randomly. and also we shuffled the data during training to help training converge fast and to prevent model from learning the order of training.

* **RMS:**

Model achieved an average accuracy of 72.34% using hyperparameters in table 3.3 on RMS optimizer. Results of best accuracy achieved shown in fig 3.42.

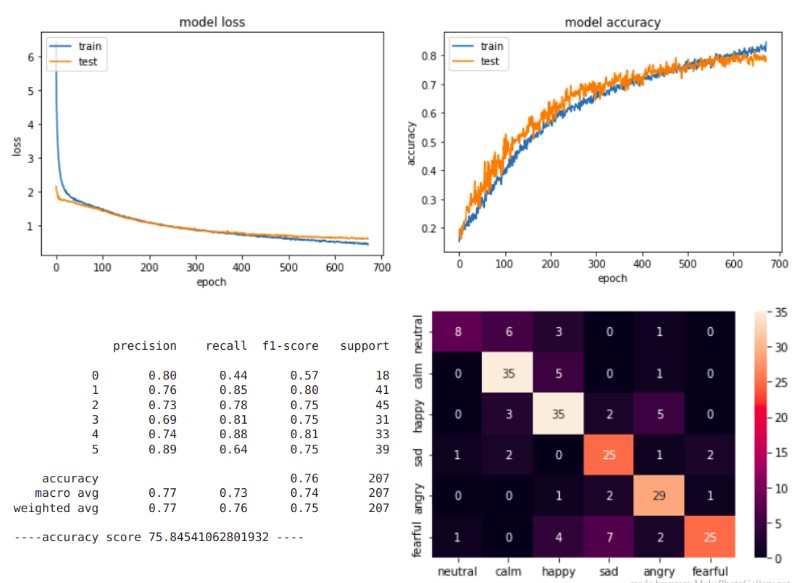


*Figure 3-42 : Shows model loss, accuracy plot, classification report and heatmap of confusion matrix*

*respectively*

**ADAM:**

Using *ADAM* optimizer on this model with hyperparameters in table 3.3 achieved average accuracy 73.14 %, Fig 3.43 shows results of best accuracy.

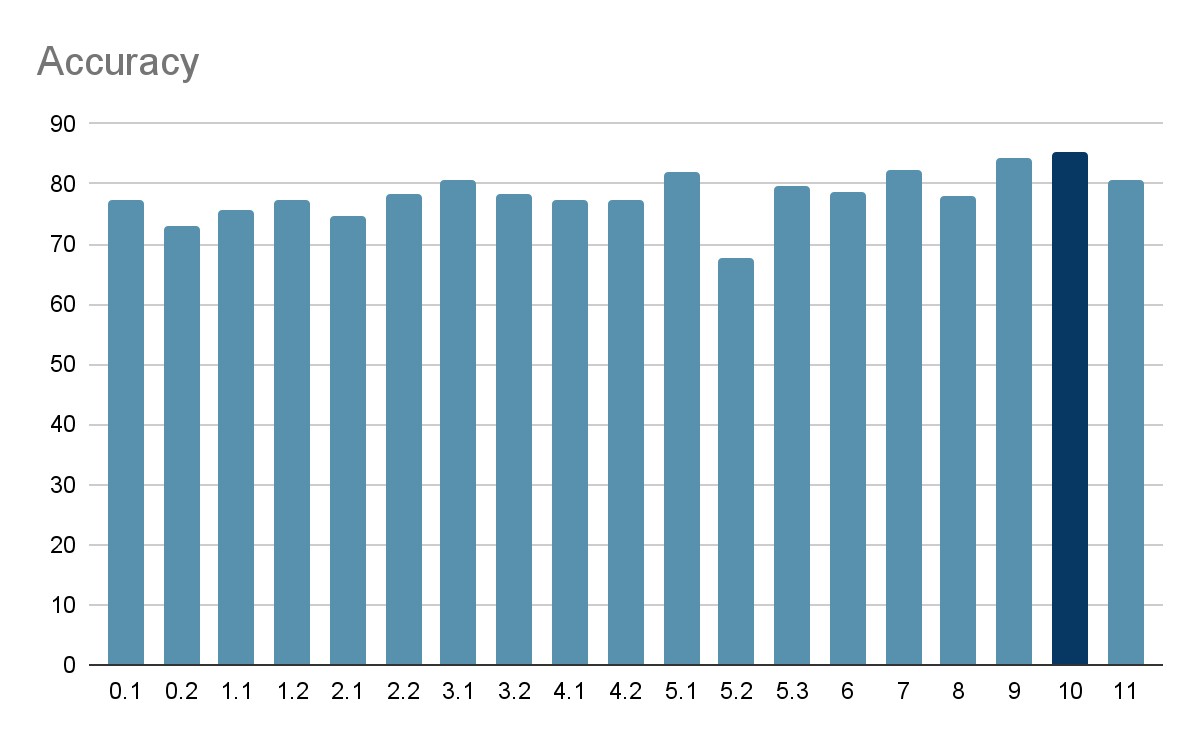


*Figure 3-43: Shows model loss, accuracy plot, classification report and heatmap of confusion matrix respectively*

* Conclusion:

##### After trying both Adam and RMS, we noticed that the difference in accuracy was in favor of Adam, but the difference was not big, the difference was (0.8%), so we will do other experiments to make sure that there is a clear and consistent difference in accuracy. 3.3.2.4 Conclusion

After making this modifications model achieved accuracy 85.25% on average and 87.43% on best



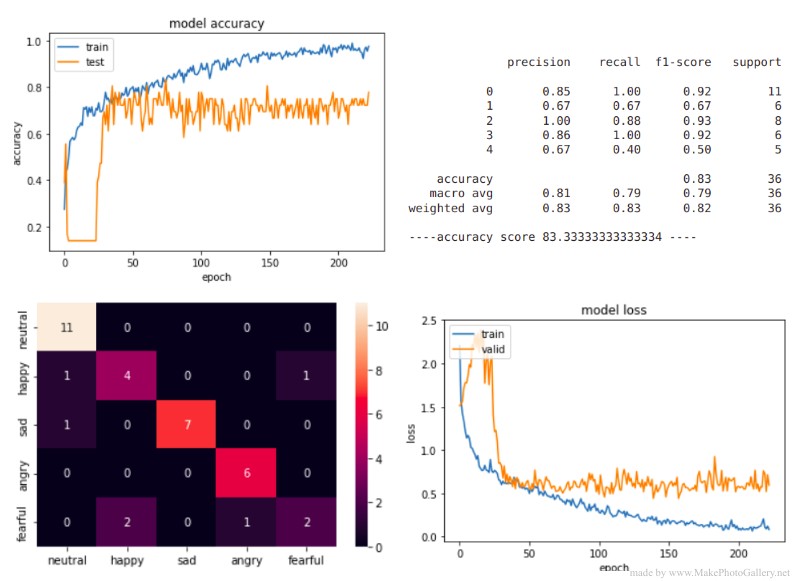
*Figure 3-61: Chart shows accuracy of modifications of CNN model*

*Table 3-4 : Show modifications on CNN model*

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **modific**  **ation** | **Dropou**  **t** | **# of**  **Dens e layer** | **Augm entati on** | **batch norm alizat ion** | **# of conv**  **layer s** | **Kern el size** | **Optimi zer** | **lr** | **Activ ation** | **# of filters** | **Accuracy** |
| 0.1 | 0.3 | 3 | NO | NO | 2 | 8 | RMS | 0.00002 | Relu | 128,256 | 77.34% |
| 0.2 | 0.3 | 3 | NO | NO | 2 | 8 | Adam | 0.00002 | Relu | 128.256 | 73.14% |
| 1.1 | 0.3 | 3 | NO | NO | 2 | 8 | RMS | 0.0002 | Relu | 128,256 | 75.55% |
| 1.2 | 0.3 | 3 | NO | NO | 2 | 8 | Adam | 0.0002 | Relu | 128,256 | 77.39% |
| 2.1 | 0.3 | 3 | NO | NO | 3 | 8 | RMS | 0.0002 | Relu | 128,256 | 74.59% |
| 2.2 | 0.3 | 3 | NO | NO | 3 | 8 | Adam | 0.0002 | Relu | 128,256 | 78.48% |
| 3.1 | 0.3 | 1 | NO | NO | 2 | 8 | RMS | 0.0002 | Relu | 128,256 | 80.57% |
| 3.2 | 0.3 | 1 | NO | NO | 2 | 8 | Adam | 0.0002 | Relu | 128,256 | 78.37% |
| 4.1 | 0.1  0.3 | 1 | NO | NO | 2 | 8 | RMS | 0.0002 | Relu | 128,256 | 77.38% |
| 4.2 | 0  0.3 | 1 | NO | NO | 2 | 8 | RMS | 0.0002 | Relu | 128,256 | 77.2% |
| 5.1 | 0.3 | 1 | NO | NO | 2 | 12 | RMS | 0.0002 | Relu | 128,256 | 82% |
| 5.2 | 0.3 | 1 | NO | NO | 2 | 4 | RMS | 0.0002 | Relu | 128,256 | 67.62% |
| 5.3 | 0.3 | 1 | NO | NO | 2 | 16 | RMS | 0.0002 | Relu | 128,256 | 79.7% |
| 6 | 0.3 | 1 | NO | NO | 2 | 12 | RMS | 0.0002 | Relu | 64,128 | 78.8% |
| 7 | 0.3 | 1 | YES | NO | 2 | 12 | RMS | 0.0002 | Relu | 128,256 | 82.2% |
| 8 | 0.1  0.3 | 1 | NO | YES | 2 | 12 | RMS | 0.0002 | Relu | 128,256 | 78.08% |
| 9 | 0.1  0.3 | 1 | NO | YES | 2 | 12 | RMS | 0.0002 | Relu | 128,256 | 84.42% |
| 10 | 0.1  0.3 | 1 | NO | YES | 2 | 12 | Adam | 0.0002 | Relu | 128,256 | 85.26% |
| 11 | 0.1  0.3 | 1 | NO | YES | 2 | 12 | Adam | 0.0002 | tanh | 128,256 | 80.79% |

**CNN MODEL ON SAVEE DATASET:**

Now we will train and test our final model on a different data set (SAVVE) that mentioned in [chapter two], the model achieved accuracy 79.16 % on average and 83.33% on best. fig 3.62 shows results of best accuracy.



*Figure 3-62: Shows model accuracy plot, classification report, heatmap of confusion matrix and loss plot respectively*

#### 3.3.4 Model Work Conclusion

After we tried a machine learning model like SVM model with different kernels and also trying multilayer perceptron (MLP) in section 3.3.3 we concluded that the highest accuracy achieved is 71.08% using SVM which is not enough to be our test accuracy and also we cannot improve the model in order to achieve a high enough accuracy to be our test accuracy. Therefore we followed deep learning algorithms. So we tried two different models, two parallel CNN (3.3.1), CNN (3.3.2) and made many modifications to them to achieve high accuracy, the outputs were as shown in table 3.6.

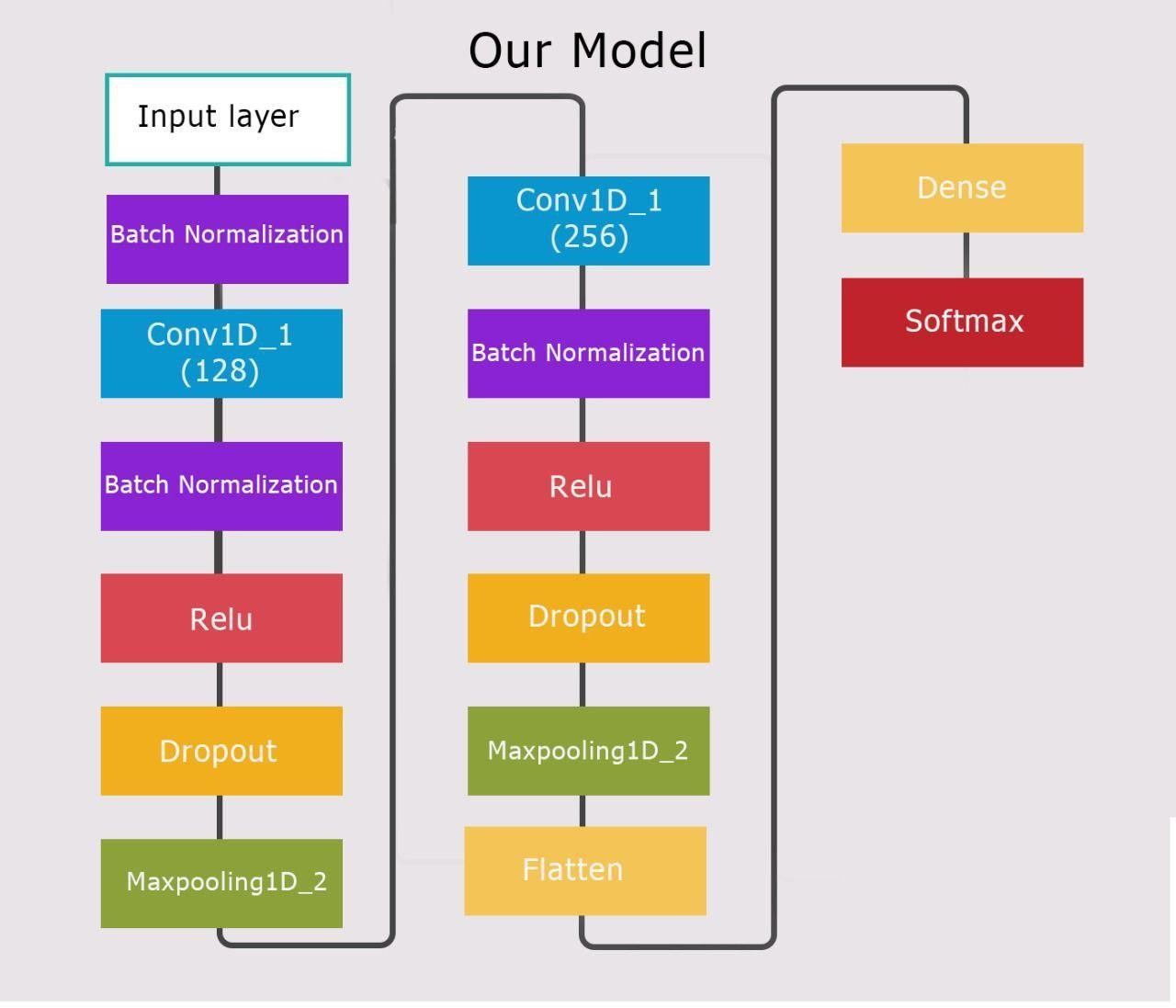
*Table 3-6 : Show accuracy comparison between reference model and our model*

|  |  |  |  |
| --- | --- | --- | --- |
| Model | Reference accuracy | Implementation of reference model | Our model |
| Two parallel  CNN | 75% | 73.23% | 80.6% |
| CNN | 70.38% | 72.15% | 87% |

From the result shown in table 3.6 we obtained that our modifications on two parallel CNN model improved accuracy by 6% approximately. and our modifications on CNN model improved accuracy by 17% approximately. We chose CNN model to be our final model due to the higher accuracy and less complication from the Two parallel CNN model.

**OUR FINAL MODEL ARCHITECTURE:**

Our final model is 2 convolution layers, First convolution layer with filter size 128 and kernel size 12 and dropout ratio 0.1 and batch normalization after it, second convolution layer with filter size 256 and kernel size 12 and dropout ratio 0.3 and batch normalization after it, and batch normalization after input layer, using 1 dense layer and adam optimizer with learning rate 0.0002 and relu activation function. Figure 3.68 shows a diagram of our final model architecture.



*Figure 3-68 : Our Final model ARCHITECTURE*

### Chapter 4 Visual Emotion Recognition System

#### 4.1 Introduction

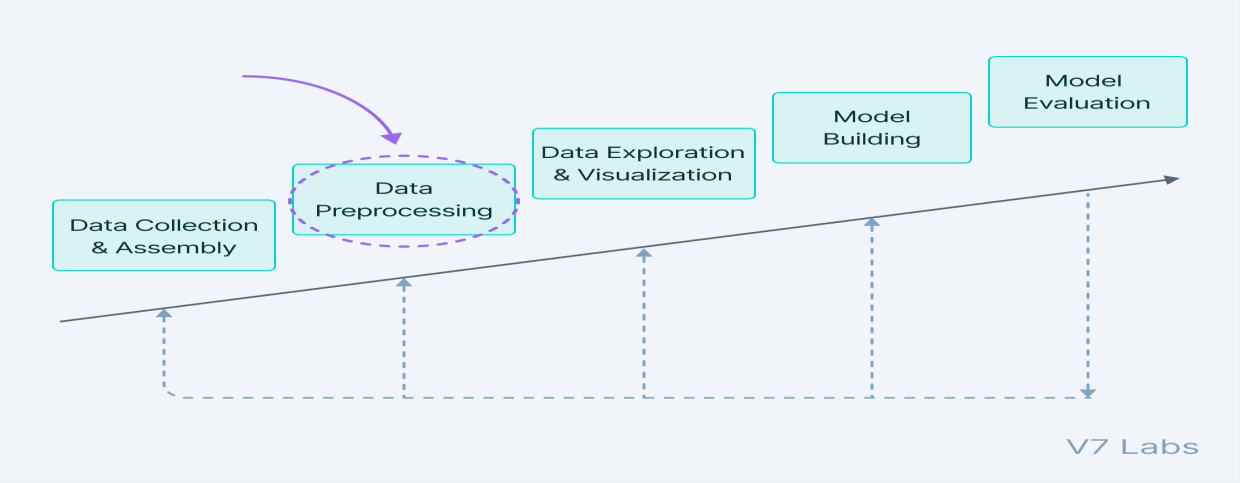
In this chapter we will explain our visual emotion recognition system. And how to deal with video data sets and how to extract features to feed into the network. We Will clearly explain our network which used to classify our emotion from videos in Detail. And we will show our results on both RAVDESS and SAVEE data sets. We Will clearly explain all these in sections in detail. In section 4.2 show how to deal with Videos and extract features. In section 4.3 explain our model's structure LRCN Model [32] and ConvLstm model [33] in detail. In section 4.4 explain ConvLstm model, How to Improve it and results on RAVDESS and SAVEE data sets. In section 4.5 Explain LRCN model, how to improve it and results on RAVDESS and SAVEE data Sets. In Section 4.6 explain our conclusion.

#### 4.2 Features Extraction

In this section the preprocessing done on data will be discussed, Section 4.2.1 discuss a general introduction about data preprocessing and why it is important Then what steps we used in our project. section 4.2.2 is a section about landmarks And the RAVDESS facial landmarks dataset that we used to process the frames. In Section 4.2.3 we discuss in details what preprocessing we done on RAVDESS Dataset, Then section 4.2.4 is about what preprocessing done on SAVEE.

##### 4.2.1 Data Preprocessing

Data Preprocessing includes the steps we need to follow to transform or Encode data so that it may be easily parsed by the machine; it is a common first step In the deep learning workflow to prepare raw data in a format that the network can Accept. For example, you can resize image input to match the size of an image input Layer. You can also preprocess data to enhance desired features or reduce artifacts That can bias the network. For example, you can normalize or remove noise from Input data[34].Common data preprocessing methods are: resizing, face detection, Cropping, adding noises, and data normalization consists of local normalization, Global contrast normalization and histogram equalization[35]. Figure 4.1 shows the Data preprocessing is an early stage in any machine learning project.



*Figure 4-1: Model training steps.*

Preprocessing methods that are used on our dataset are:

1. Divide videos into labels representing the emotions.
2. Divide each video into frames and label them.
3. Crop each frame using landmarks.
4. Resize each frame into 64\*64.
5. Normalized each frame.
6. Convert labels for each video to one hot encoded.

##### 4.2.2 Landmarks

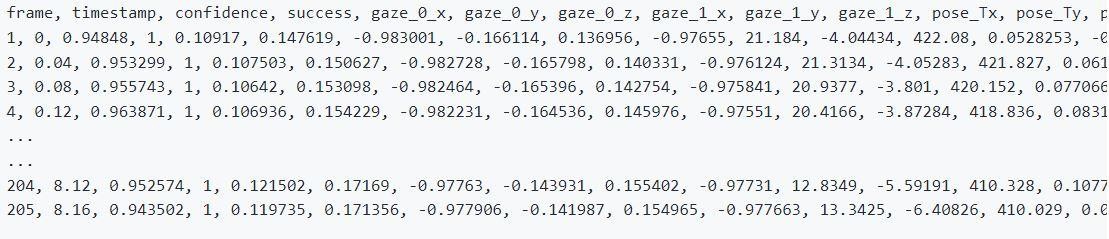
Facial landmark detection is the task of detecting key landmarks on the face and

Tracking them. Facial landmark detection algorithms help to automatically identify The

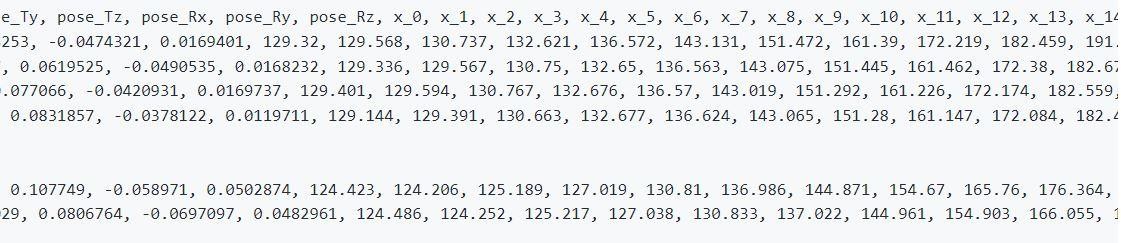
Locations of the facial key landmark points on a facial image or from a video. The key Landmark points normally include the facial regions like nose tip, eye Corner, Eyebrows and chin tip. Some applications of facial landmark detection are Face Swap, head pose detection, detecting facial gestures, gaze direction [36].

**RAVDESS Landmark Dataset**

The Landmarks used to crop frames in the RAVDESS database. Landmark is A dataset containing tracked facial landmark movements. This dataset is found in [37][38] .Motion tracking of actors' faces was produced by OpenFace 2.1.0. Tracked Information includes: facial landmark detection, head pose estimation, facial action Unit recognition, and eyegaze estimation.This data set contains tracking for all 2452 RAVDESS trials. All tracking movement data are contained in 2452 CSV files. Each Actor has 104 tracked trials (60 speech, 44 song). Note, there are no song files for Actor 18.Total Tracked Files = (24 Actors x 60 Speech trials) + (23 Actors x 44 Song Trials) = 2452 files. Tracking results for each trial are provided as individual comma Separated value files (CSV format). File naming convention of tracked files is Identical to that of the RAVDESS. For example, the tracked file "01-01-01-01-01-01-01.csv" corresponds to RAVDESS audio-video file "01-01-01-01-01-01-01.mp4". Fig 4.2 and 4.3 are a sample of that dataset from 01-01-01-01-01-01-01.csv file, each column in this file represents a specific Landmark which is shown in table 4.1.



*Figure 4-2: Sample of 01-01-01-01-01-01-01.csv file.*



*Figure 4-3: Sample of 01-01-01-01-01-01-01.csv file.*

*Table 4-1: Content of the csv file for facial landmarks.*

|  |  |
| --- | --- |
| **Columns** | **Represents** |
| Columns 1-3 | Timing and Detection Confidence |
| Columns 4-291 | Eye Gaze Detection |
| Columns 292-297 | Head pose |
| Columns 298-433 | Facial Landmarks locations in 2D |
| Columns 434-637 | Facial Landmarks locations in 3D |
| Columns 638-677 | Rigid and non-rigid shape parameters |
| Columns 687-712 | Facial Action Units[5] |

##### 4.2.3 Preprocessing On RAVDESS

Data preprocessing steps performed on RAVDESS are discussed in this Section. First is labeling and how we classified the dataset by labels in 4.2.3.1. Second in 4.2.3.2 frames are extracted from videos, then in section 4.2.3.3 is cropping these Frames with landmarks, lastly performing one hot encoding in section 4.2.3.4.

###### 4.2.3.1 Labeling

Instead of having 24 folders (one for each actor), each contains 60 speech Videos and 44 song videos (except for actor 18), we found it would be easier and Better to classify these videos into only 6 folders (neutral, happy, angry, calm, fearful And sad), each one represents an emotion (which are the 6 emotions that are Common between speech and song) and contains all videos that belongs to that Emotion from all actors.

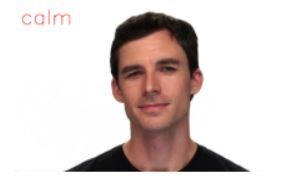
Now each folder contains (8 no. of emotion videos x 24 actors =192 speech video) + (8 no. of emotion videos x 23 actors= 184 song video) = 376 videos.

Neutral is an exception as each actor has only 8 neutral video (4 for speech & 4 for

Song), therefore, the total number of videos in neutral folder is (4 no. of emotion Videos x 24 actors =96 speech videos) + (4 no. of emotion videos x 23 actors= 92 Song video ) = 188 videos.

###### 4.2.3.2 Frames

Each video is then divided into frames, 30 frames are taken from each video. 30 frames from each video will then be fed to the model as one sequence. A frame Shown in fig 4.4 for Actor1 with calm emotion.



*Figure 4-4: Sample of Actor1 frame.*

###### 4.2.3.3 Cropping With Landmarks

The facial Landmarks dataset mentioned above was then used to crop each One of these frames so that only the face of the actor is focused at and the rest of The Background is neglected. This is shown in fig 4.5 that shows a frame for actor1 After Being cropped. each frame is then resized to a height and width equal 64.



*Figure 4-5: Cropped Actor1 frame.*

###### 4.2.3.4 One Hot Encoding

It is a common way of preprocessing categorical features for machine learning Models. This type of encoding creates a new binary feature for each possible Category and assigns a value of 1 to the feature of each sample that corresponds to Its original category. In our project we used Keras's to\_categorical method to convert Labels into one-hot-encoded vectors.[39]

##### 4.2.4 Preprocessing On SAVEE

Data preprocessing steps performed on SAVEE are discussed in this section. First is labeling and how we classified the dataset by labels in 4.2.4.1. Second in 4.2.4.2 Frames are extracted from videos, lastly performing one hot encoding in section 4.2.4.3.

###### 4.2.4.1 Labeling

Instead of having 4 folders (one for each actor), each contains 90 speech Videos , we found it would be easier and better to classify these videos into only 5 Folders (neutral, happy, angry, fearful and sad), each one represents an Emotion(which are the 5 emotions that are found common with RAVDESS) and Contains all videos that belongs to that emotion from all actors.

Now each folder contains (15 no. of emotion videos x 4 actors =60 speech videos).

Neutral is an exception as each actor has 30 neutral videos, therefore , the total Number of videos in the neutral folder is (30 no. of emotion videos x 4 actors =120 Speech videos).

###### 4.2.4.2 Frames

Each video is then divided into frames, 30 frames are taken from each video, Each frame is resized to a height and width equal 64.30 frames from each video will Then be fed to the model as one sequence. Same as done on RAVDESS.A frame Shown in fig 4.6 for KL with sad emotion.



###### 4.2.4.3 One Hot Encoding

The same hot encoding was done here on the SAVEE data set as Keras's To\_categorical Method is used to convert labels into one-hot-encoded vectors.

##### 4.2.5 Dataset Splitting

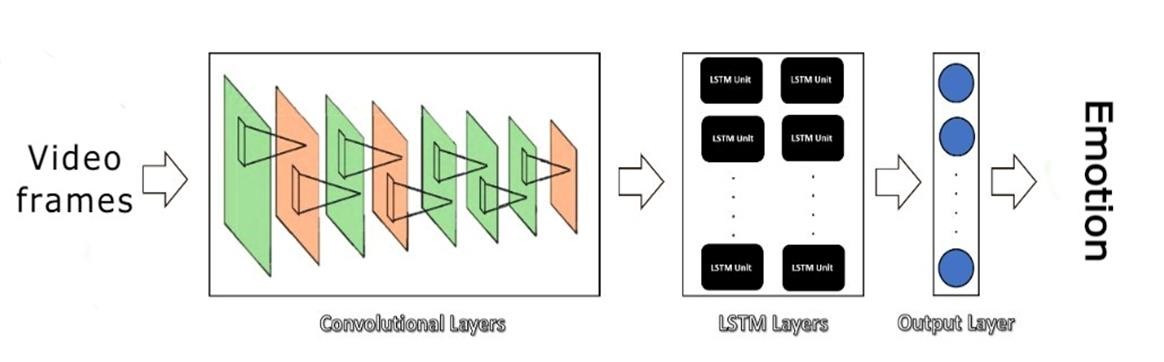
Data splitting is commonly used in machine learning to split data into a train, Test, or validation set. This approach allows us to find the model hyper-parameter and also estimate the generalization performance. Our dataset was at first split to 80% for training and 20% for testing, and the validation is split as 20% of the Training data. After this we tried another split which resulted in better performance which is 90% for training and 10% for test, and the validation is 10 % of the training.

#### 4.3 LRCN & ConvLstm Structure Model

In this section we will talk about our LRCN and ConvLstm models structure And run original LRCN and ConvLstm on RAVDESS and SAVEE datasets. Section 4.3.1 shows the LRCN model structure. Section 4.3.1.1 shows LRCN architecture Implementation. Section 4.3.1.2 shows LRCN model results on the RAVDESS dataset And SAVEE dataset. Section 4.3.2 shows the ConvLstm model structure. Section 4.3.2.1 shows LRCN architecture implementation. Section 4.3.2.2 shows LRCN Model results on the RAVDESS dataset and SAVEE dataset.

##### 4.3.1 LRCN Model

LRCN (Long-Recurrent Convolutional Network) approach combines Convolution and LSTM layers in a single model. Another approach can be to use CNN model and LSTM model trained separately. But in this approach we combine CNN and LSTM in one model. The Convolutional layers are used for spatial feature Extraction from the Video frames, and the extracted spatial features are fed to LSTM Layer(s) at each timestep for temporal sequence modeling. This way the network learns spatiotemporal features directly in an end-to-end training, resulting in a Robust Model. Figure 4.7 shows the LRCN network which takes as an input video Frames and outputs the emotion or classification.



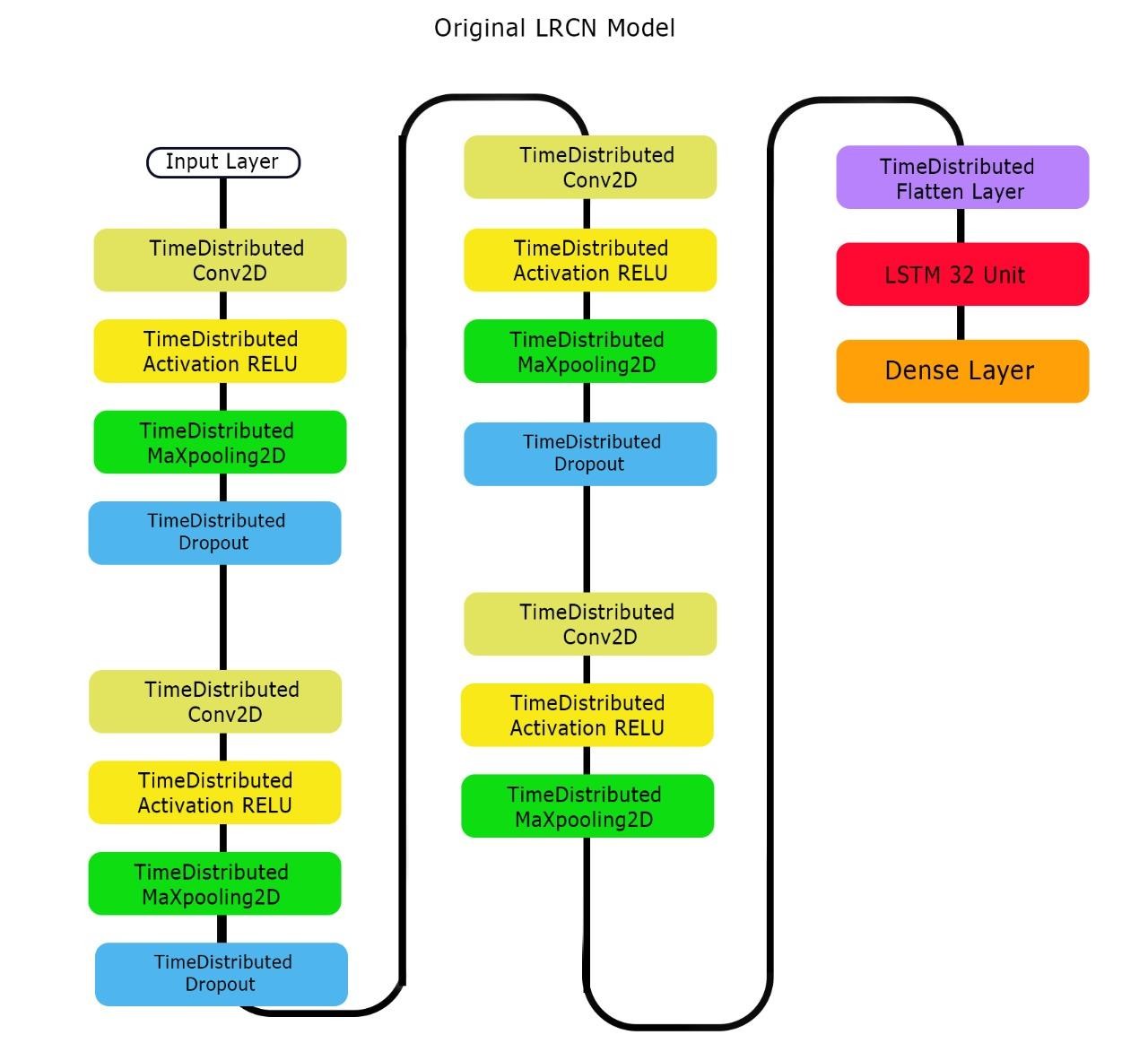
*Figure 4-7: LRCN network.*

This LRCN architecture is introduced or proposed by the Long-term Recurrent

Convolutional Networks for Visual Recognition and this LRCN model used in “” Longterm Recurrent Convolutional Networks for Visual Recognition and Description “”paper[32]. We used a Time Distributed wrapped layer so that every frame of video Can be applied to the same layer independently. So it makes a layer capable of Taking input of shape (No of frames, width, height, No of channels). if the input Shape was (width, height, No of channels) it allows to input the whole video into the Model in a single shot. In our work we used input shape (No of frames, width, height, No of channels). In our work with LRCN model we used 30 frame and 64\*64 (width \* Height) as mentioned in the preprocessing section and No of channels equals 3.

###### 4.3.1.1 LRCN Architecture Implementation

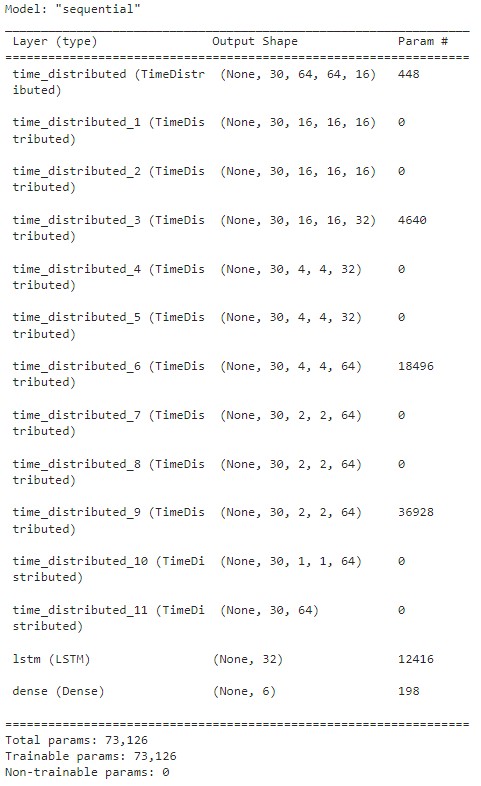
To implement our LRCN architecture, we will use time-distributed **Conv2D** Layers which will be followed by **MaxPooling2D** and **Dropout** layers. The feature Extracted from the **Conv2D** layers will be then flattened using the **Flatten** layer and will be fed to a **LSTM** layer. The **Dense** layer with Softmax activation will then use the output from the **LSTM** layer to predict the emotion. Our model diagram is shown in figure 4.8.



*Figure 4-8: Original LRCN model structure.*

###### 4.3.1.2 LRCN Model Results

In this section we will show the results when applying the LRCN model on the RAVDESS dataset and SAVEE dataset. Figure 4.9shows the model report and Number of parameters and trainable parameters in the LRCN model.

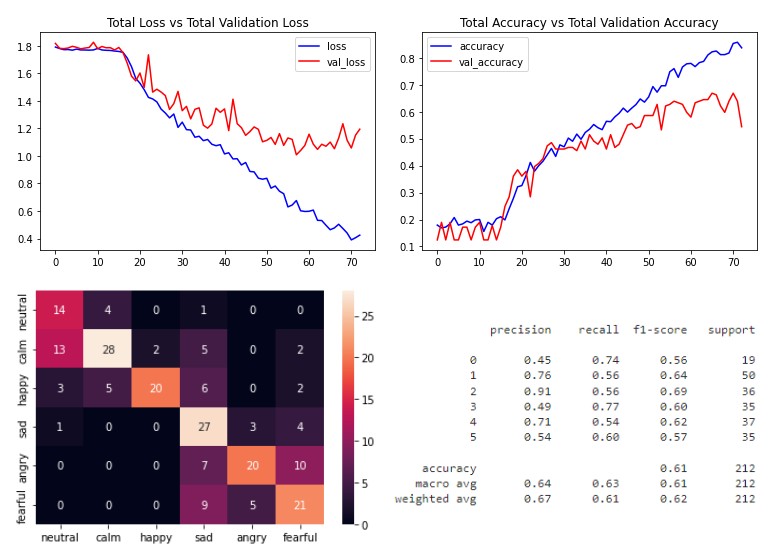


*Figure 4-9: Original LRCN model structure report.*

**RAVDESS Dataset**

Speech

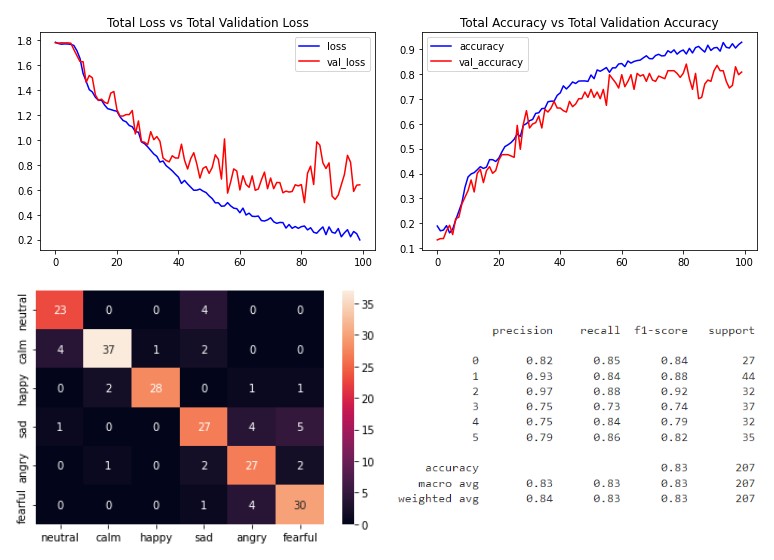
We first split the dataset into a train and test set with test size 20%.Then Validation split from the train set 20%. We trained the model with optimizer Adam and categorical\_crossentropy as a loss function which is used for multi-class Classification models. We trained the LRCN model with these hyper parameters. Learning rate used is the default for the optimizer Adam in Kera’s which is 0.001, Number of epochs equals 90 with early stopping function to prevent overfitting. The Model stopped training at epoch number 73. The LRCN model achieved 61.32%. We Show loss and accuracy graph, heatmap of confusion matrix and classification report In figure 4.10.



*Figure 4-10: Results of original LRCN model on RAVDESS speech only.*

Speech and Song

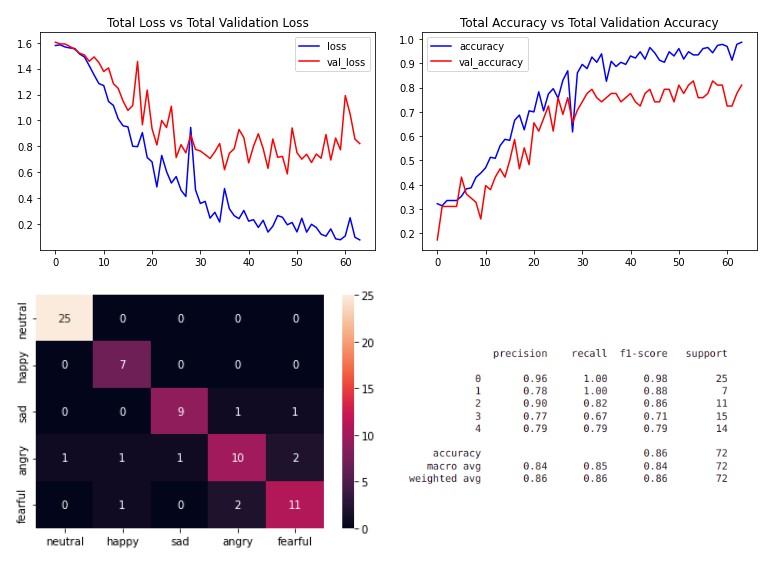
We first split the dataset into a train and test set with test size 10%.Then Validation split from the train set 10%. We trained the model with optimizer Adam And categorical\_crossentropy as a loss function which is used for multi-class Classification models. We trained the LRCN model with these hyperparameters. Learning rate used is the default for the optimizer Adam in Keras which is 0.001, Number of epochs equals 100 with early stopping function to prevent overfitting. But The model is trained with all epochs instead of an early stopping function. The LRCN Model achieved 83.09% as the best accuracy. And achieved 80.67% ± 2.18%. We Show loss and accuracy graph, heatmap of confusion matrix and classification report in figure 4.11.



*Figure 4-11: Results of original LRCN model on RAVDESS.*

SAVEE Dataset

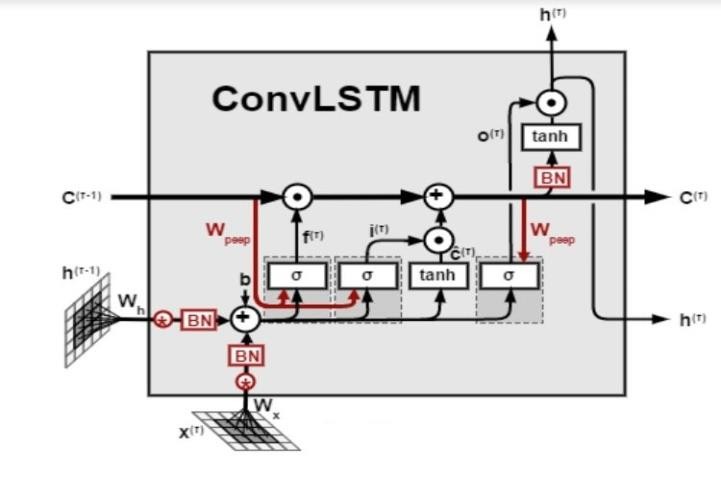
We first split the dataset into a train and test set with test size 20%.Then Validation split from the train set 20%. We trained the model with optimizer Adam And categorical\_crossentropy as a loss function which is used for multi-class Classification models. We trained the LRCN model with these hyperparameters. Learning rate used is the default for the optimizer Adam in Keras which is 0.001, Number of epochs equals 100 with early stopping function to prevent overfitting. But The model is trained with all epochs instead of an early stopping function. The LRCN Model achieved 90.28% as the best accuracy.The LRCN model achieved 87% ± 2.78%.we present loss and accuracy graph, heatmap of confusion matrix and Classification report in figure 4.12.



*Figure 4-12: Results of original LRCN on SAVEE*

##### 4.3.2 ConvLstm Model Approach

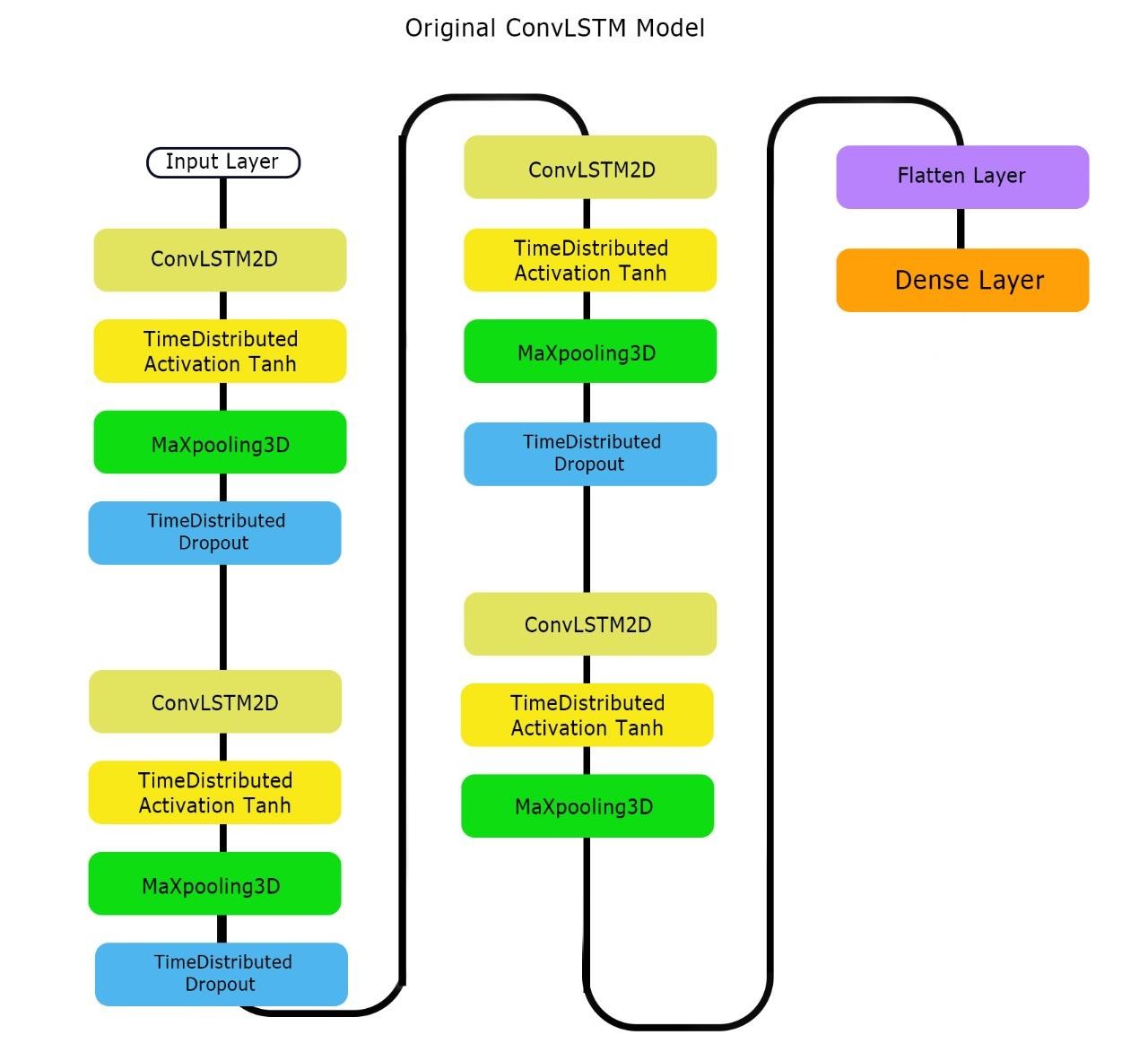
We will implement the first approach by using a combination of ConvLstm Cells. A ConvLstm cell is a variant of an Lstm network that contains convolutions Operations in the network. It is an Lstm with convolution embedded in the Architecture, which makes it capable of identifying spatial features of the data while Keeping into account the temporal relation. And this ConvLstm model approach is Used in “Convolutional Lstm Network: A Machine Learning Approach for Precipitation Nowcasting “paper [33]. Show ConvLstm cell in figure 4.13.



*Figure 4-13: ConvLstm cell.*

###### 4.3.2.1 ConvLstm Model Implementation

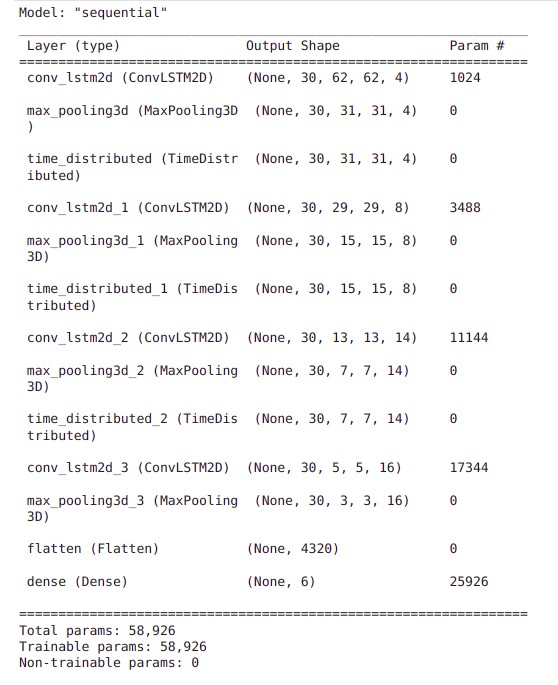
To construct the model, we will use Keras **ConvLstm2D** recurrent layers. The **ConvLstm2D** layer also takes in the number of filters and kernel size required For Applying the convolutional operations. The output of the layers is flattened in the end and is fed to the **Dense** layer with softmax activation which outputs the Probability of each action category. We will also use **MaxPooling3D** layers to reduce the dimensions of the frames and avoid unnecessary computations and **Dropout** layers to prevent overfitting the Model on the data. The architecture is a simple one and has a small number of Trainable parameters. This is because we are only dealing with a small subset of the Dataset which does not require a large-scale model. Our model diagram is shown in Figure 4.14.



*Figure 4-14: Original ConvLstm structure*

###### 4.3.2.2 ConvLstm Model Results

In this section we will show the results when applying the ConvLstm model On The RAVDESS dataset. First we will show the results of the model on RAVDESS Dataset speech only then we will show the results using speech & song dataset. Figure4.15shows the ConvLstm model report and number of parameters and Trainable parameters in the ConvLstm model.



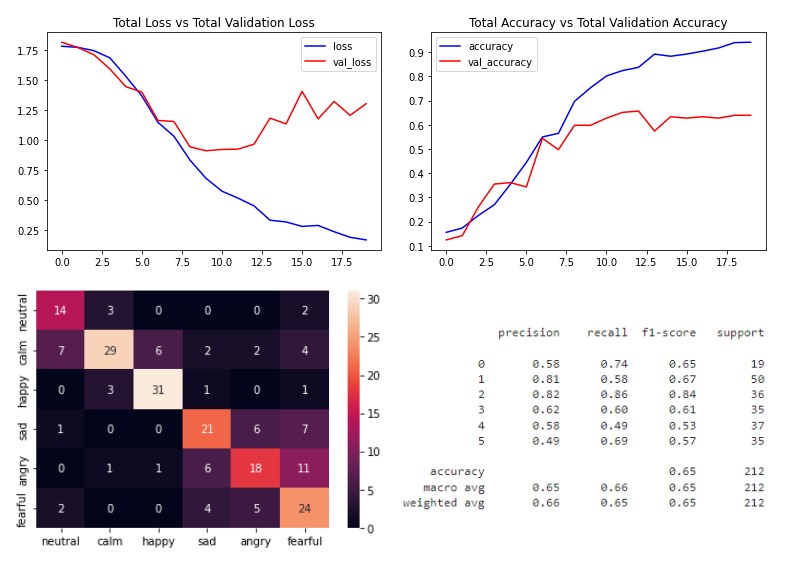
*Figure 4-15: Original ConvLstm model report.*

RAVDESS Dataset

We split the dataset to 60% train 20% validation data 20% test data, set hyperparameter learning rate is default(0.001) ,optimizer is Adam.

Speech

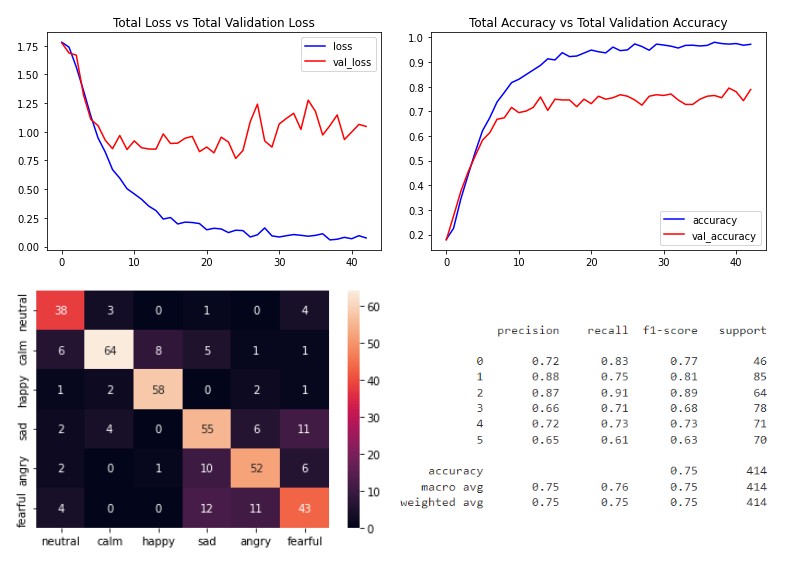
It achieved accuracy 64.62%. Loss and accuracy graphs, heatmap of Confusion matrix and classification report show in figure 4.16.



*Figure 4-16: Results of Original ConvLstm model on RAVDESS speech.*

Speech and Song

It achieved accuracy 74.8%. Loss and accuracy graphs, heatmap of confusion Matrix and classification report show in figure 4.17.

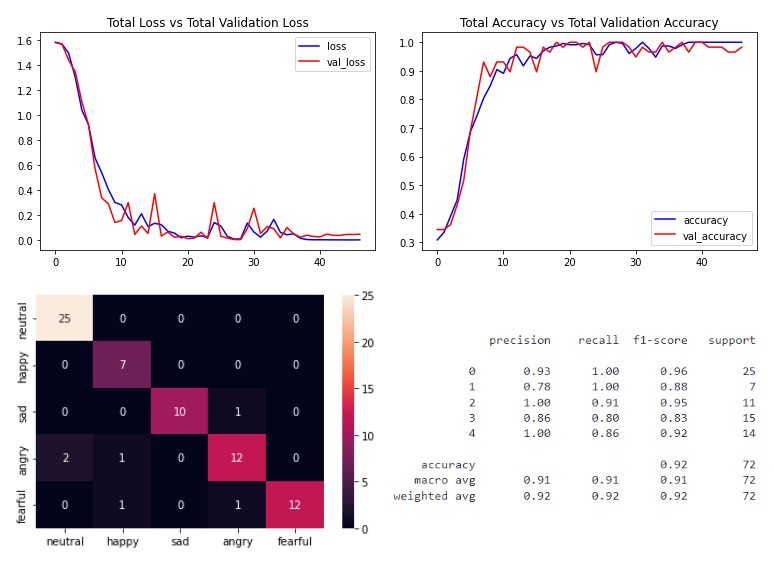


*Figure 4-17: Result of Original ConvLstm model on RAVDESS.*

SAVEE Dataset

We first split the dataset into a train and test set with test size 20%.Then

Validation split from the train set 20%. The learning rate is default(0.001), optimizer Is Adam, number of epochs equals 70 with early stopping function to prevent Overfitting. The model is stopped at epoch number 29. The ConvLstm model Achieved 91.67%. We present loss and accuracy graph, heatmap of confusion Matrix and classification report in figure 4.18.

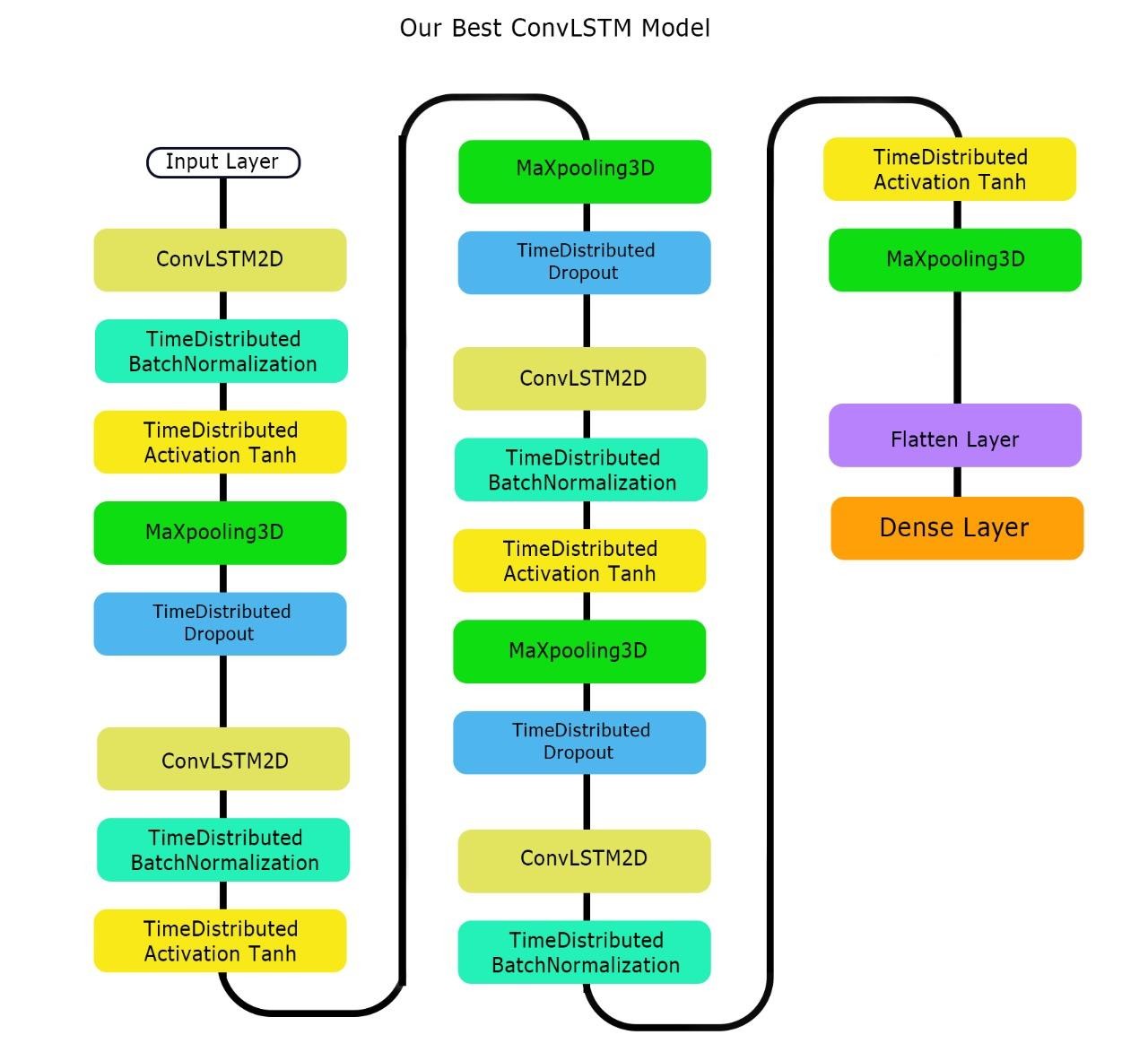


*Figure 4-18: Results of original ConvLstm model on SAVEE.*

##### 4.4.2 Results of final ConvLstm Model

Our best model is ConvLstm model [33] with adding batch normalization, dropout rate is 0.2. This Model

Achieved best accuracy on a RAVDESS data set 84.54% and on a SAVEE Data set 94.44%. We will show our best ConvLstm model in figure 4.20. We will Show graphs of loss, accuracy, heatmap of confusion matrix and classification report of RAVDESS And SAVEE results on our best ConvLstm model in figures 4.21.



*Figure 4-20: Final ConvLstm model.*

##### 4.5.6 Results Of Final LRCN Model

Our Final Model is LRCN model [32] with adding batch normalization, drop out rate is 0.1:0.2:0.3, # of LSTM units equal 64 and batch size equal 5. This model Achieved Best accuracy on a RAVDESS data set 96.62% and on a SAVEE data set 97.22%. We will find that SAVEE has a higher accuracy, but that is because of some Reasons. We found that the fact that there are only four actors in this database does Not help And the model doesn’t receive a great amount of variety.

Although the Video files in The test set are not included in the training set, the actors expressing That emotion Have. Therefore every face has already been seen by the trained Model and Therefore some of the facial characteristics will have already been saved And learnt. In [40], they split the dataset in two different ways.

At first they split it the Normal way As they separated the files by emotions, and the testing accuracy was Very high (95%), the second way was separating the files by actors, as all files of One actor Went into the test set and the rest went to the training and validation, a 21% Accuracy is obtained which is way below the first try.

This proves that the model Was Not learning but remembering certain characteristics of the actors’ faces. We Can Observe that the actors in the SAVEE dataset have blue marks painted on their Faces. This could be a reason for the poor results since what the model could be Doing is remembering the positioning of these blue marks for every actor for the Different emotions.

Hence, if the model has never seen the actor before, then the Model will fail to recognize the emotions. All these things explain why the SAVEE Dataset has a higher accuracy. We will show hyperparameters of the original and Final LRCN model in table 4.8, the final LRCN model diagram in figure 4.26 and Graphs of Loss, Accuracy, heatmap of confusion matrix and classification report of RAVDESS And SAVEE results on our best LRCN model in figures 4.27.

*Table 4-8: Hyperparameter of original and final LRCN model.*

|  |  |  |
| --- | --- | --- |
| **Hyper-parameter** | **Original LRCN** | **Final LRCN MODEL** |
| Optimizer | Adam | Adam |
| # Epochs | 70 | 100 |
| Learning rate | Default | Default |
| Dropout rate | 0.25 | 0.1 |
| 0.25 | 0.2 |
| 0.25 | 0.3 |



*Figure 4-26: Final LRCN model.*

#### 4.6 Conclusion

In our project we used two models LRCN and ConvLstm which used to Classify human activity in “Long-term Recurrent Convolutional Networks for Visual Recognition and Description” and “Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting” [32][33]. Original ConvLstm model Achieved on RAVDESS data set 74.8% and on a SAVEE dataset 91.67%. The final ConvLstm was achieved on the RAVDESS data set 84.54% and on the SAVEE Dataset 94.44%. Original LRCN model was achieved on a RAVDESS data set 80.67% ± 2.18% and on a SAVEE data set 86.81% ± 2.78%. The final LRCN model Was achieved on a RAVDESS data set 93.27% ± 2.66%. And on SAVEE data set 95.24% ± 2.78%. We will find that the final LRCN model achieved an accepted Accuracy rather than our final ConvLstm model in both RAVDESS and SAVEE data Sets. Our final visual emotion recognition model is LRCN model with adding batch Normalization, dropout rate is 0.1:0.2:0.3, # of LSTM units equal 64 and batch size equal 5.

#### Chapter 5 Conclusion & Future work

##### 5.1 Conclusion

Our project consists of 2 phases: audio, video In each phase we used different techniques and tried to improve other related work accuracy.

In audio Phase, first we extract audio features using two methods Mel Spectrogram and MFCC. We tried different models using Mel spectrogram and MFCC and found that the range of accuracy of both methods are 58% - 69% and 69% - 73% respectively. we found that MFCC has higher accuracy than Mel spectrogram, so we choose MFCC to extract features from our audio files. After extracting features by MFCC, which feed into our network. We worked on two models and tried to improve them. The first model consists of two parallel CNN [28] and achieved 73% as an average accuracy. We tried to improve it and our modifications on it improved its accuracy by 7% Approximately. Our improved model achieved 80.86% as the best accuracy on the RAVDESS dataset. The second model, we implement the model architecture in “Emotion Recognition from Speech Signals Using Machine Learning and Deep Learning Techniques” paper [10], Original model achieved 70.83% on the RAVDESS dataset and classified 8 emotions. Our modifications On CNN model improved accuracy by 17% approximately. Our model achieved 87.4% as the best accuracy on the RAVDESS dataset and classified 6 emotions. And 83.3% as the best accuracy on SAVEE dataset.

*Table 5-1* : *Show our results on audio phase.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Paper** | **Year** | **Dataset** | **Paper accuracy** | **Our accuracy** |
| Emotion Recognition from Speech Signals  Using Machine Learning and Deep  Learning Techniques [10] | 2021 | RAVDESS | 70.38%  (8 emotions) | 87%  (6 emotions ) |
| SAVEE | 61.46% | 83.3 % |

In the Video phase we used landmarks to extract frames from video. We Used two models LRCN and ConvLSTM which used to Classify human activity in Two papers [32][33]. The original ConvLstm model Achieved on RAVDESS data set 74.8% and on a

SAVEE dataset 91.67%. Our Modifications on ConvLSTM model improved accuracy by 10% approximately on RAVDESS. The final ConvLstm was achieved on the RAVDESS data set 84.54% and On the SAVEE Dataset 94.44%.

The second model LRCN, the original LRCN model was achieved on the RAVDESS data Set 80.67% ± 2.18% and on the SAVEE data set 86.81% ± 2.78%. Our modifications On LRCN model improved accuracy by 13% approximately on the RAVDESS dataset. The final LRCN model Was achieved on the RAVDESS data set 93.27% ± 2.66%. And on the SAVEE data set 95.24% ± 2.78%.

##### 5.2 Future work

We will deploy our audio and visual system in different applications. such as an **Emotion recognition app for online admissions and interviews.** which can be used to understand how candidates feel during interviews and how they react to certain questions. This information Can be used to optimize interview structure for future candidates and Streamline the application process. **Emotion analysis for an online education app.** which can be used for online education is an ideal way to analyze the Online student journey and improve it where necessary. Assess Schools course materials, teaching styles, structure and layout by way of emotional feedback as student’s go through each module in Real-time. **Emotion analysis for marketing.** which can be used by different companies to gauge consumer Mood towards their products, brands, marketing efforts, staff or In-location experiences. Understanding customer emotions is vital to Ensure business growth and enhance experiences. **Automotive industry and emotion analysis.** Car manufacturers around the world are increasingly focusing on Making cars more personal and safe for people to drive. Using facial Emotion detection smart cars can alert the driver when he is feeling Drowsy and in turn help to decrease road casualties. In Research The model can also be improved further to improve its quality through New modifications to the model to get a better accuracy, and the quality of the Current model can be used in transfer learning in different fields.

##### ABBREVIATIONS

|  |  |
| --- | --- |
| **ABBREVIATION** | **SENTENCE** |
| AO | audio overlapping |
| MFCC | Mel-frequency cepstral coefficients |
| CNN | convolutional neural network |
| RAVDESS | The Ryerson Audio-Visual Database of Emotional Speech and Song |
| SAVEE | Surrey Audio-Visual Expressed Emotion |
| WA | weighted accuracy |
| MLP | Multi layer perceptron |
| RNN | Recurrent neural network |
| LRCN | Long-term recurrent convolutional network |
| ConvLstm | Convolutional Long ShortTerm Memory |
| Lr | learning rate |
| Opt | optimizer |
| LR | Logistic Regression |
| SVM | Support Vector Machine |
| MLP | Multilayer Perceptron |
| KNN | K-Nearest Neighbor |
| Crema-d | Crowd-sourced Emotional multi-modal Actors Dataset |
| MRPN | multi-modal Residual Perceptron Network |

**References**

1. **Audio Analysis & Machine Learning**
   * G. Mendels, "Towards Data Science," 18 Nov 2019. [Online]. Available: https://towardsdatascience.com/how-to-apply-machine-learning-and-deep-learning-methods-to-audio-analysis-615e286fcbbc.
2. **Video Preprocessing for Emotion Recognition**
   * L. T. Alberto Albiol, "Video preprocessing for audiovisual indexing," *IEEE Computer Society Annual Symposium on VLSI*, p. 4, 2014.
3. **Speech & Video Emotion Detection**
   * N. Ellarby Sánchez, "Emotion Detection through Audio and Video," 2021.
   * N. E. S. S. M. A. C. Q. F. E. Moncholi, *Final Degree Thesis: "Emotion Detection through Audio and Video,"* June 2021.
4. **Audio Representation in Machine Learning**
   * A. H. L. Lu, L. M. T. Ö., and E. B., "Audio Representation," MA: Springer US, p. 8, 2009.
5. **Spectrograms for Speech Emotion Recognition**
   * Wikipedia contributors, "Spectrogram," *Wikipedia*, 10 February 2022. [Online]. Available: <https://en.wikipedia.org/w/index.php?title=Spectrogram&oldid=1070997200>.
6. **Speech Emotion Recognition Metrics**
   * A. Bajaj, "Performance Metrics in Machine Learning - Complete Guide," *Neptune AI Blog*, 4 January 2022. [Online]. Available: https://neptune.ai/blog/performance-metrics-in-machine-learning-completeguide.
7. **Speech Emotion Recognition Datasets**
   * S. R. Livingstone and F. A. Russo, "The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English," *PloS One*, 2018.
   * P. Jackson and S. Haq, "Surrey Audio-Visual Expressed Emotion (SAVEE) Database," 2014.
8. **Deep Learning for Speech Emotion Recognition**
   * J. A. R. and W. B. Abdul Malik Badshah, "Speech Emotion Recognition from Spectrograms with Deep Convolutional Neural Network," *International Conference on Platform Technology and Service (PlatCon)*, February 2017.
   * O. S. A. Mohamed, "Arabic Speech Emotion Recognition Employing Wav2vec2.0 and HuBERT Based on BAVED Dataset," 2021.
   * A. Aftab et al., "Light-SERNet: A Lightweight Fully Convolutional Neural Network for Speech Emotion Recognition," 2021.
9. **Time-based Speech Emotion Recognition**
   * B. Puterka and J. Kacur, "Time Window Analysis for Automatic Speech Emotion Recognition," *IEEE*, p. 4, 16-19 September 2018.
10. **Video-based Facial Emotion Recognition**

* N. E. B. Hajarolasvadi and H. D. Hajarolasvadi, "Video-based Person-dependent and Person-independent Facial Emotion Recognition," pp. 1049-1056, 2021.
* S. Zhang et al., "Learning Affective Features with a Hybrid Deep Model for Audio-Visual Emotion Recognition," *Transactions on Circuits and Systems for Video Technology*, 2017.

1. **GitHub Repositories for Emotion Recognition**

* Lina Zenkov, "Transformer CNN Emotion Recognition," GitHub, 6 Nov 2020. [Online]. Available: <https://github.com/IliaZenkov/transformer-cnn-emotion-recognition>.
* Kosta-jo, "Speech Emotion Classification with PyTorch," GitHub, 31 March 2021. [Online]. Available: <https://github.com/Data-Science-kosta/Speech-Emotion-Classification-with-PyTorch>.

1. **Deep Learning Frameworks & Preprocessing**

* MathWorks, "Deep Learning Data Management and Preprocessing," [Online]. Available: https://www.mathworks.com/help/deeplearning/deep-learning-data-management-and-preprocessing.html.