FEATURE EXTRACTION:

The initial stage of a speech recognition system is to extract features or identify the components of the audio signal that are useful for detecting linguistic content while ignoring everything else, such as background noise, emotion, and so on. mfcc (Mel Frequency Cepstral Coefficient), chroma, and mel (Mel Spectrogram Frequency) are all crucial elements from the sound file, as is appending the values in a data frame. For 1D CNN, chroma, MFCC, and Spectrogram were employed, whereas, for 2D CNN, Spectrogram and data augmentation were used.

Data Augmentation - AWGN:

Additive White Gaussian noise (AWGN) is a fundamental noise model used in information theory to simulate the effects of numerous random processes seen in nature. Because it is added to any noise that may be inherent in the information system, it is additive. The term "white" alludes to the concept that the information system has homogeneous power across the frequency spectrum. It's like the colour white, which emits the same amount of light at all frequencies in the visible spectrum. Because it has a normal distribution in the time domain and an average time-domain value of zero, it is referred to be Gaussian.

Convolutional Neural Network (CNN):

A Convolutional Neural Network is a Deep Learning system from the domain chosen to various aspects of the domain chosen by assigning relevance and distinguishing between them. When compared to other classification methods, the amount of pre-processing required by a ConvNet is significantly less. While basic approaches need hand-engineering of filters, ConvNets can learn these filters/characteristics with enough training. A number of similar fields can be stacked on top of each other to span the full visual field.

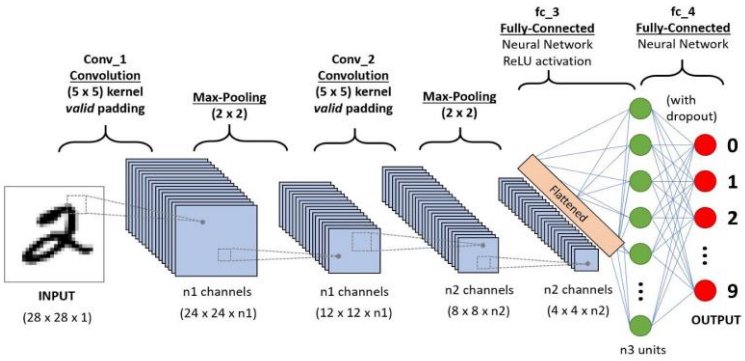

Fig 3.1 Convolutional Neural Network

Long Short-Term Memory (LSTM):

Long Short-Term Memory networks, or "LSTMs," are a kind of RNN that can learn long-term dependencies. The control flow of an LSTM is comparable to that of a recurrent neural network. It processes data and passes the information on as it moves along. The processes within the cells of the LSTM vary. All recurrent neural networks are made up of a series of repeated neural network modules. This repeating module in ordinary RNNs will have a relatively basic structure, such as a single tanh layer. LSTMs have a chain-like structure as well, but the repeating module is different. Instead of a single neural network layer, there are four, each of which interacts in a unique way.

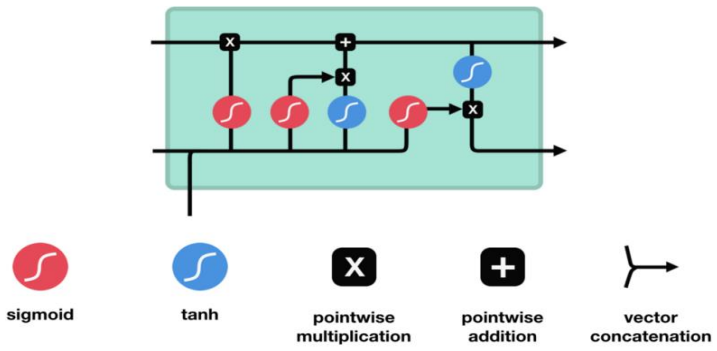

Fig 3.2 Long Short-Term Memory

The cell state and its many gates are at the heart of LSTM. The cell state serves as a transportation route for relative information as it travels down the sequence chain. In principle, the cell state can carry meaningful information throughout the sequence's

processing. As a result, information from earlier time steps might make its way to later time steps, lessening the short-term memory effects.

The LSTM may delete or add information to the cell state, which is carefully controlled by structures called gates. The gates are several neural networks that determine whether information about the cell state is permitted. During training, the gates might learn what information is important to preserve or forget.

Bi-Directional Long Short-Term Memory (Bi-LSTM):

Bidirectional LSTMs are a kind of LSTM that can help improve model performance when dealing with sequence classification problems. When all time steps of the input sequence are known, LSTM will train only one input sequence whereas in Bi-directional LSTM trains both the way. This results in one of the input sequences being the replica of the other one. This can provide additional context for the network, allowing it to comprehend the problem more quickly and comprehensively.

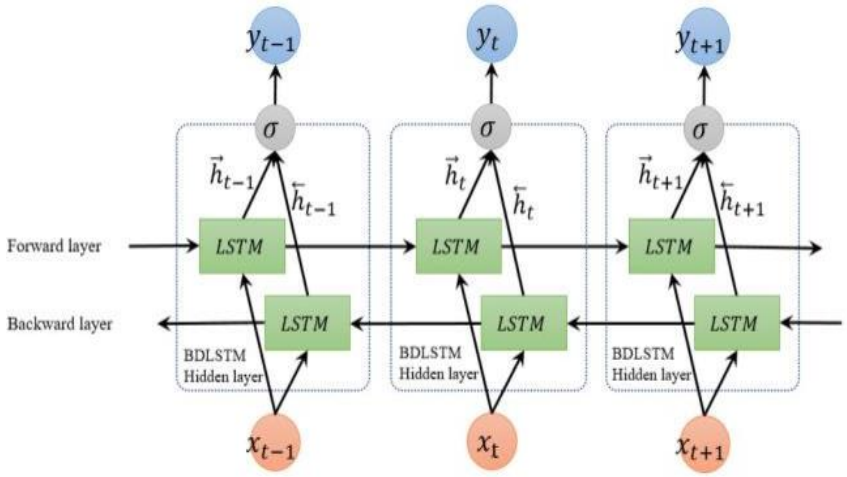

Fig 3.3 Bi-Directional Long Short-Term Memory

EXPERIMENTAL RESULTS

DATASET:

The RAVDESS comprises speech audio-only files (16bit, 48kHz.wav), there are 1440 files in this section of the RAVDESS: 60 trials per actor multiplied by 24 actors equals 1440 trials. The RAVDESS is a group of 24 professional actors (12 women, 12 men) who perform two lexically-related phrases in a neutral North American dialect. Calm, joyful, sad, furious, afraid, surprise, and disgust are examples of spoken emotions. Each expression has two emotional intensity levels (normal and strong), as well as a neutral expression. Every one of the 1440 files has its own filename. A seven-part numerical identifier makes up the filename (e.g., 03-01-06-01-02-01-12.wav).

1-Dimensional CNN:

The model is constructed in a way that it has 4 conv1d layers with the activation layers and Batch Normalization layer. This layer normalizes its inputs. Batch normalization is a transformation that keeps the mean output near 0 and the standard deviation of the output close to 1.

During training and inference, batch normalization operates in several ways. During training, the layer uses the mean and standard deviation of the current batch of inputs to normalize its output. The layer normalizes its output during inference by taking a moving average of the mean and standard deviation of the batches it saw during training.

Layer (type)	Output Shape	Param #
conv1d_6 (Conv1D)	(None, 180, 256)	1536
activation_7 (Activation)	(None, 180, 256)	0
conv1d_7 (Conv1D)	(None, 180, 128)	163968
batch_normalization_3 (Batch	(None, 180, 128)	512
activation_8 (Activation)	(None, 180, 128)	0
dropout_2 (Dropout)	(None, 180, 128)	0
max_pooling1d_1 (MaxPooling1	(None, 22, 128)	0
conv1d_8 (Conv1D)	(None, 22, 128)	82048
batch_normalization_4 (Batch	(None, 22, 128)	512
activation_9 (Activation)	(None, 22, 128)	0
conv1d_9 (Conv1D)	(None, 22, 128)	82048

Fig 4.1 1-D CNN layers description

Stacked Time Distributed 2D CNN – LSTM:

A TimeDistributed layer enables a layer to be applied to each temporal slice of an input. The MEL spectrogram is computed and utilized as an input to the model, which is divided into seven parts. The mel spectrogram is split into seven sections and run through four TimeDistributed Conv2D layers. Flattening reduces file size by combining all visible layers into a backdrop layer. After that, it goes via the LSTM layer. Finally, the forward pass is used to acquire the cross-entropy loss by feeding the model's logits to the softmax layer, and prediction bypassing the logits to the softmax layer.

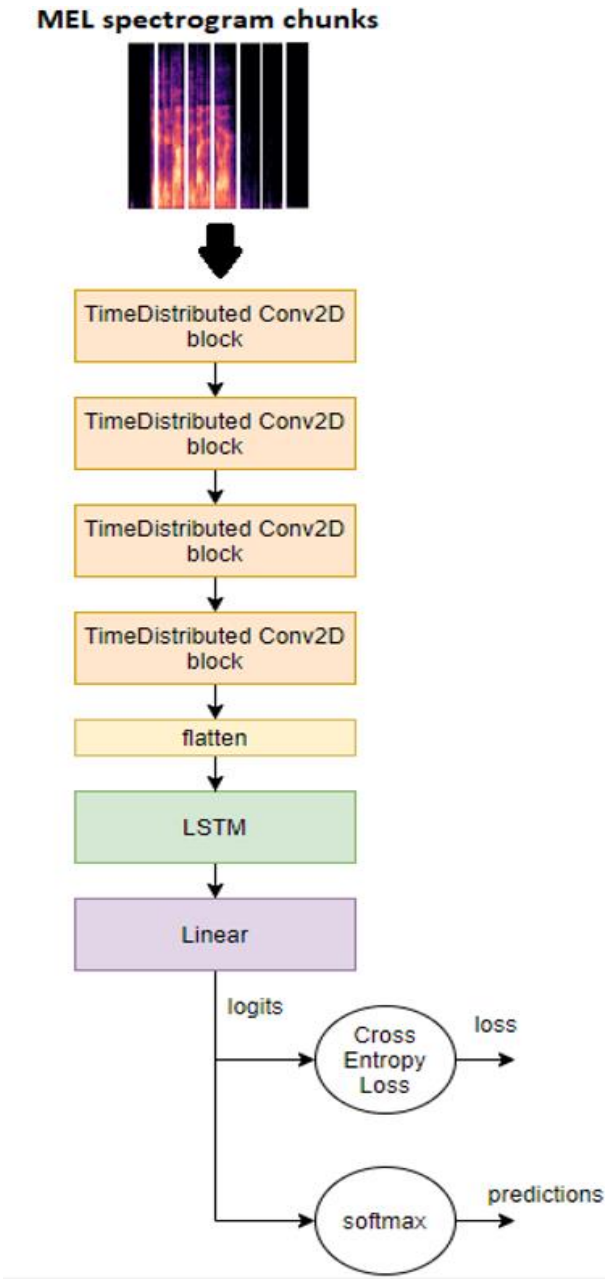

Fig 4.2 Flow chart of Stacked CNN and LSTM

Stacked Time Distributed 2D CNN – Bi-LSTM:

In addition to the Time Distributed 2D CNN - LSTM discussed previously, here the Bi-Directional = True in the LSTM layer is implemented, and the weights from Bi-LSTM are passed to the attention layer, after which the softmax normalized weights are sent to forward pass, and the loss and prediction are obtained similarly to Time Distributed 2D CNN - LSTM.

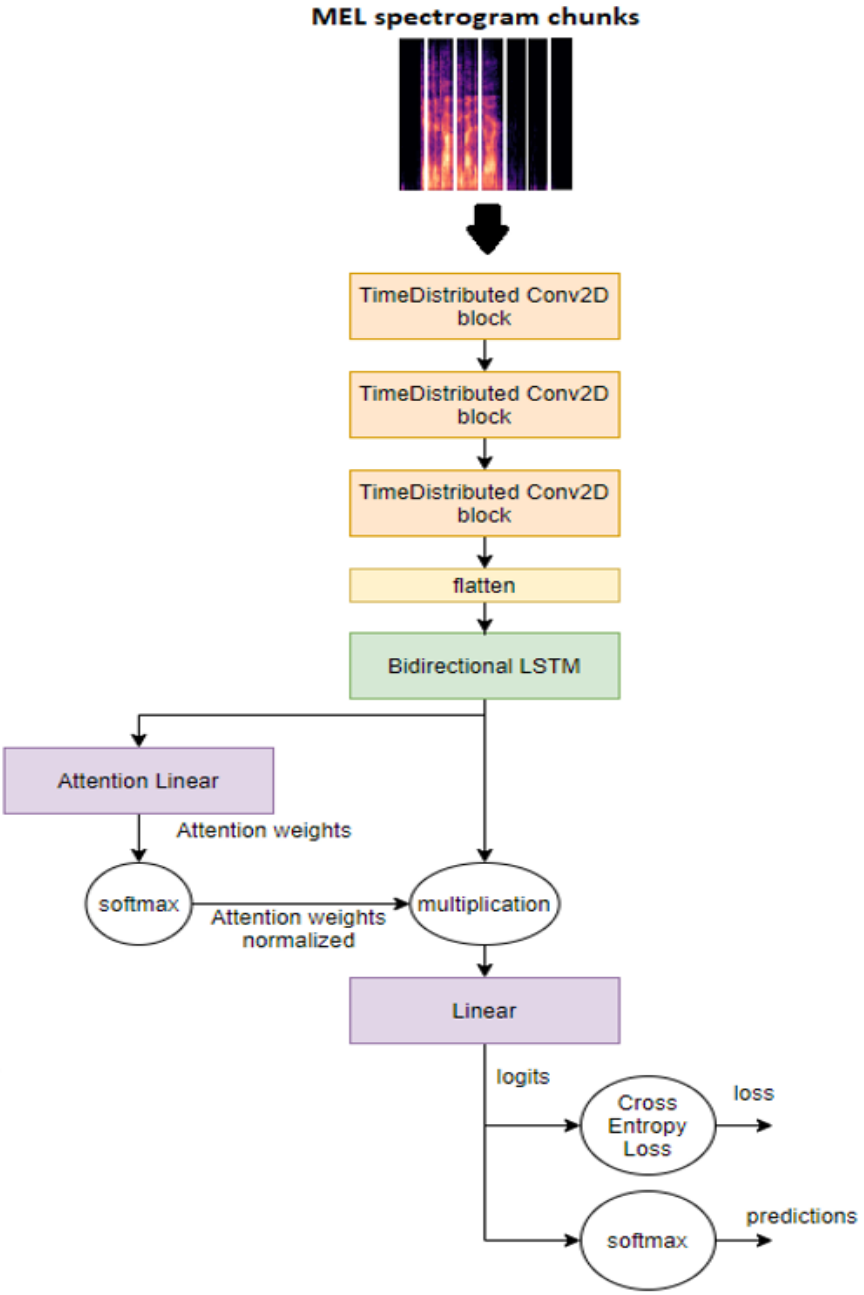

Fig 4.3 Flow chart of Stacked CNN and Bi-LSTM with attention layer

Parallel 2D CNN – Bi-LSTM:

As previously mentioned, the Time Distributed layer is not employed here; instead, the mel spectrogram feature is used. That is, after processing, both Conv 2D and Bi-LSTM with the attention layer are concatenated and delivered to forward pass.

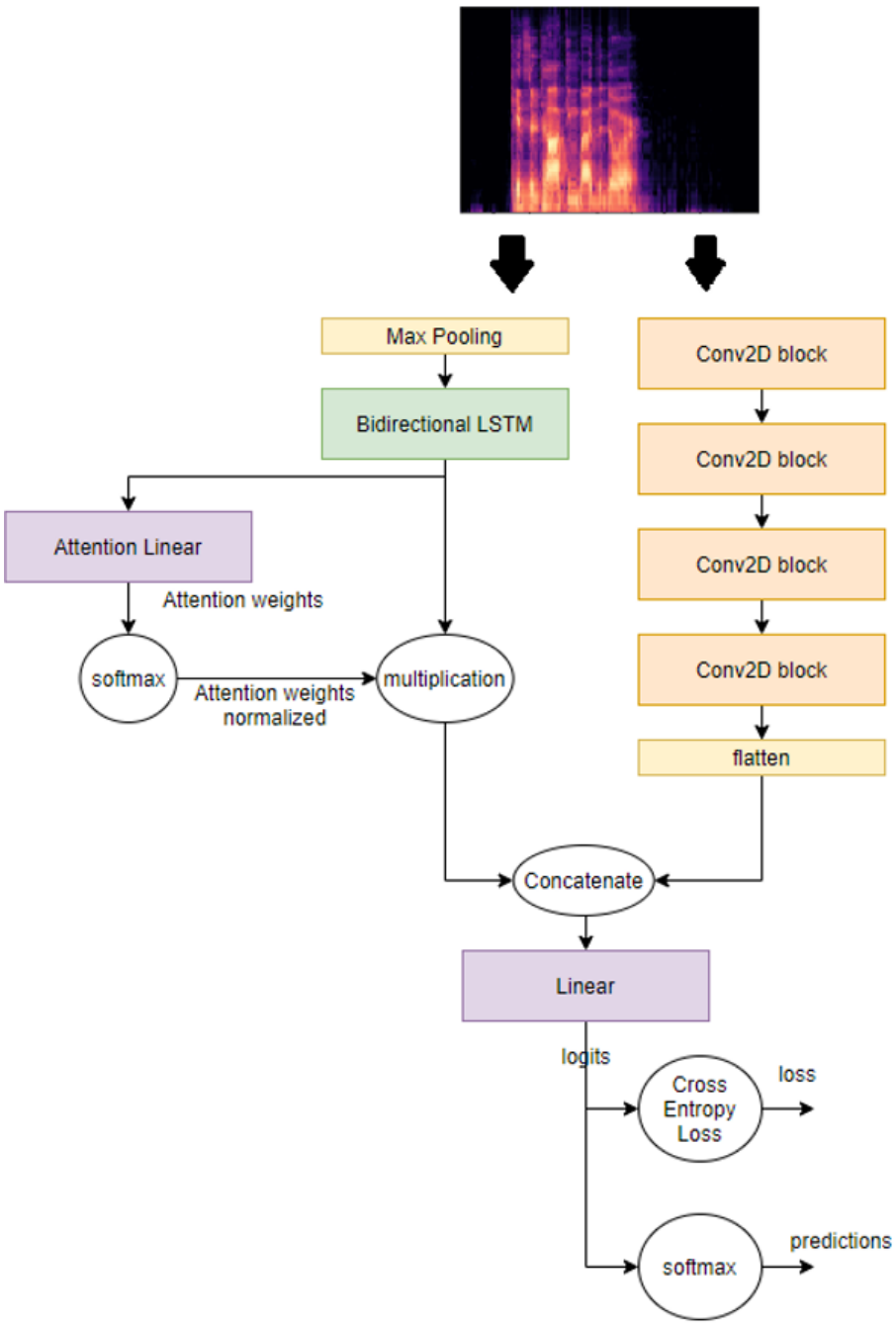

Fig 4.4 Flow chart of Parallel CNN and Bi-LSTM with attention layer

Parallel 2D CNN – Transformer Encoder:

The Transformer Encoder layer is replaced with the Bi-LSTM layer in the aforementioned model. Transformer Encoder Layer is made up of self-attn and feedforward networks. As a result, the mean of the time dimension is concatenated with the Conv 2D layer, which is then passed to the forward pass for computation loss and prediction as a whole.

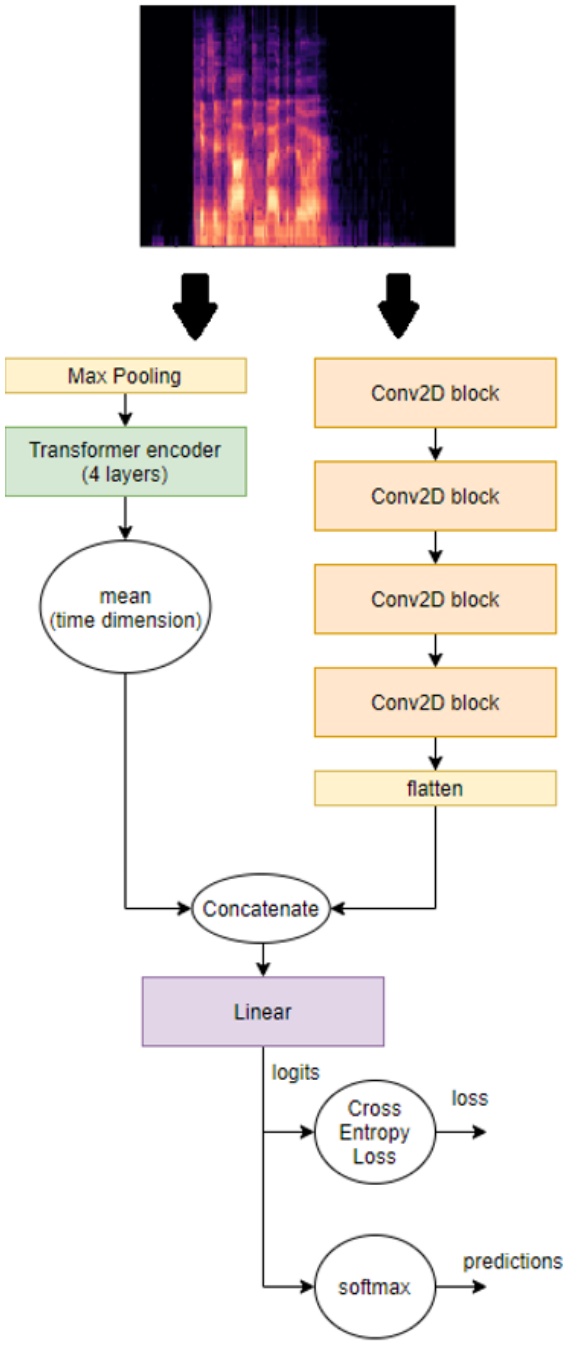

Fig 4.5 Flow chart of Parallel CNN and Transformer Encoder

MODELS ACCURACY:

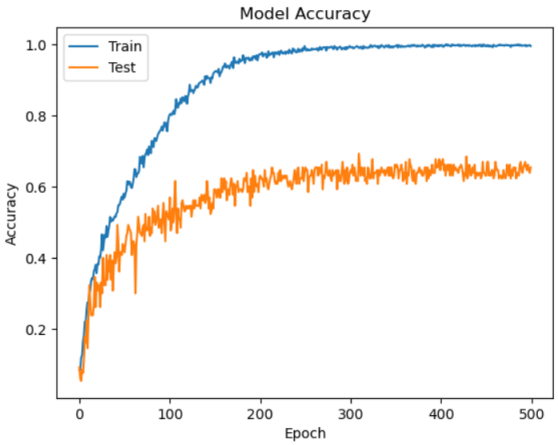

Fig 4.6 Simple CNN

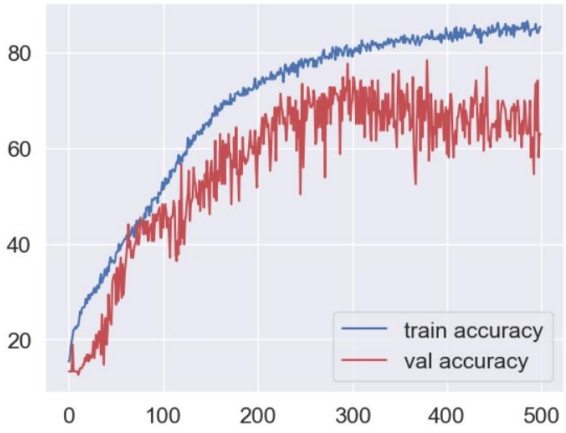

Fig 4.7 Stacked CNN and LSTM

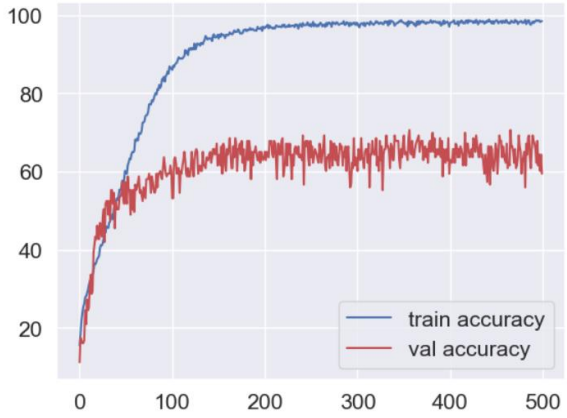

Fig 4.8 Stacked CNN and Bi-LSTM

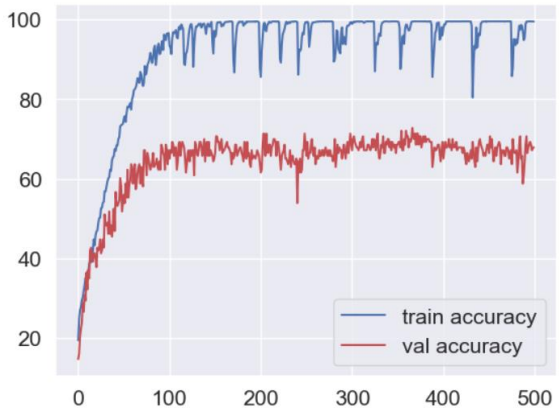

Fig 4.9 Parallel CNN and Bi-LSTM

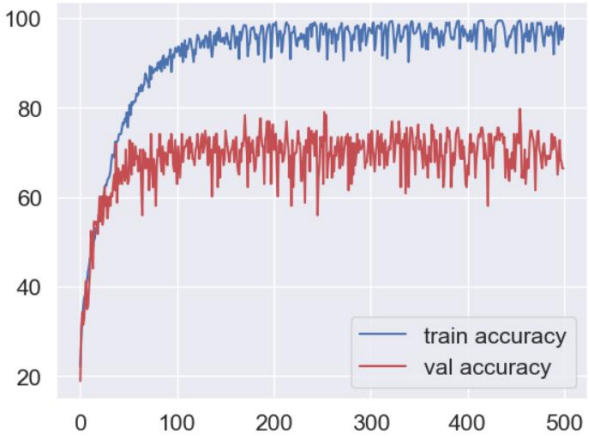

Fig 4.10 Parallel CNN and Transformer Encoder

MODELS LOSS:

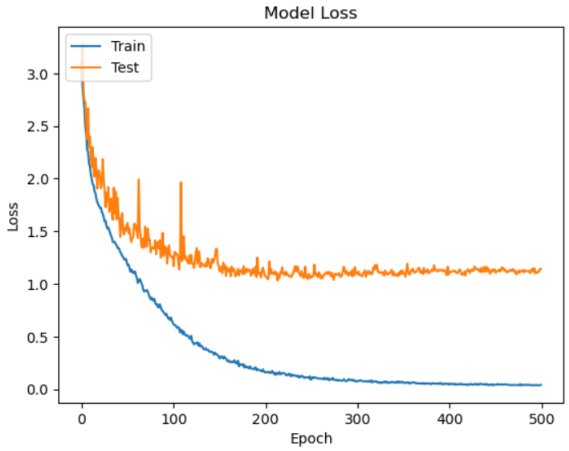

Fig 4.11Simple CNN

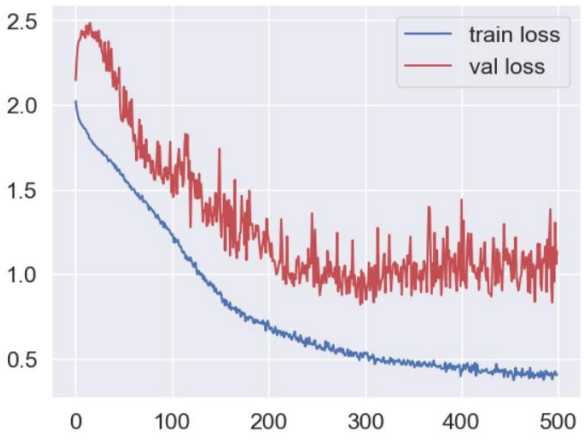

Fig 4.12 Stacked CNN and LSTM

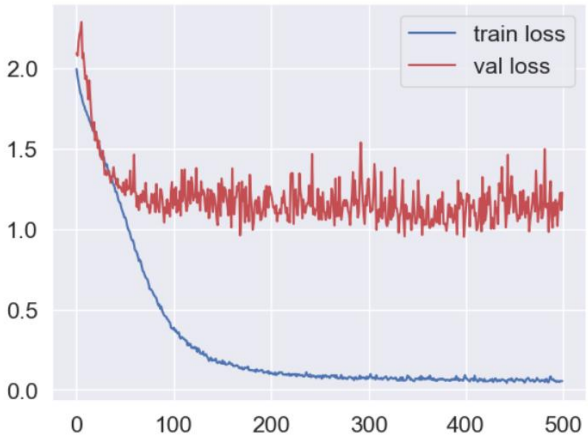

Fig 4.13 Stacked CNN and Bi-LSTM

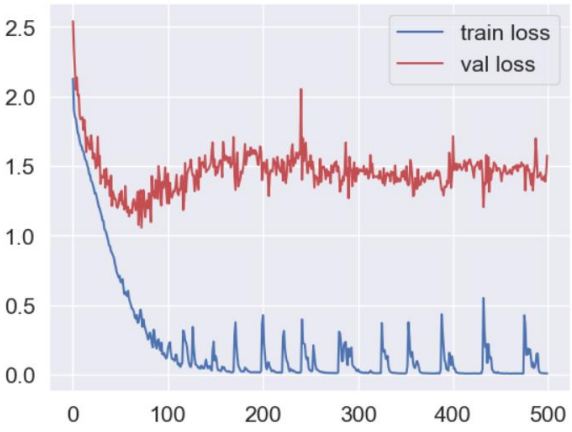

Fig 4.14 Parallel CNN and Bi-LSTM

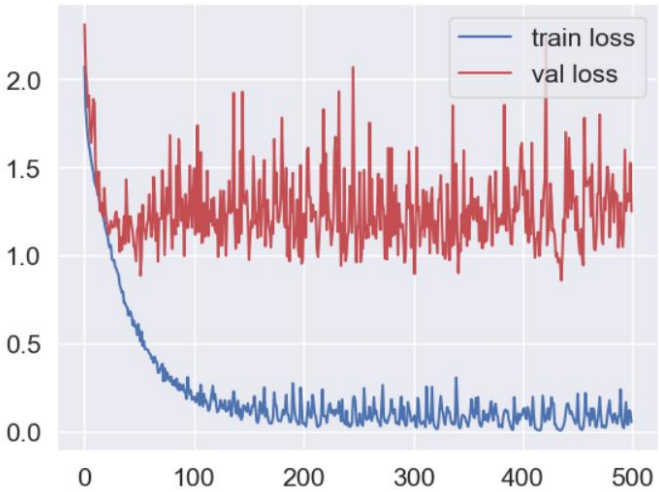

Fig 4.15 Parallel CNN and Transformer Encoder

INFERENCE:

Thus, the model is trained over,

Device: Cuda

Epochs: 500

Batch Size: 32

Loss: Categorical_Crossentropy

Optimizer: Stochastic Gradient Descent (SGD)

Learning Rate: 0.01

	1D CNN	Stack - 2D CNN & LSTM	Stack - 2D CNN & BiLSTM	Parallel - 2D CNN & BiLSTM	Parallel - 2D CNN & Transformer Encoder
Train Accuracy	99.49	85.32	98.49	99.45	97.79
Val Accuracy	65.38	62.94	59.44	67.83	66.43
Test Accuracy	70.14	57.33	54.67	62.00	70.00
Train Loss	0.42	4.059	0.537	0.110	0.540
Val Loss	11.434	11.315	12.268	15.759	12.497
Test Loss	9.7	12.49	13.46	12.78	14.79

Table 4.1 Final Results from various models

According to the following table, 1D CNN outperforms all other models over 500 epochs, although it's worth noting that there is no difference after 250 epochs. To solve this, adopting CNN + Bi-LSTM with data augmentation and attempting to increase the number of epochs parallel CNN and Transformer Encoder may produce better results than 1D CNN, as it now sits next to 1D CNN in many circumstances. Thus, among the other models, Parallel CNN with Transformer Encoder is the best