DL MINI PROJECT FINAL REPORT

MOHD ARHAM SHAIKH
200968051
BATCH - 3

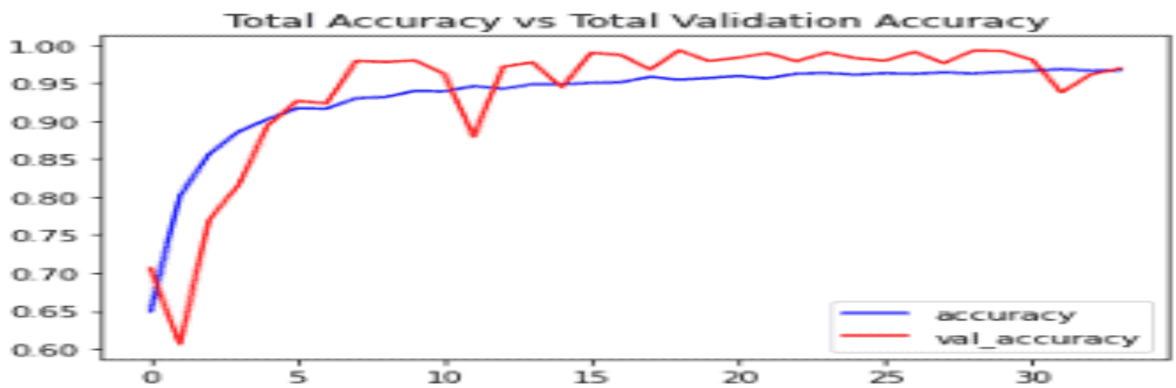
1)Tabulation and visualization of results in terms of performance and accuracy.

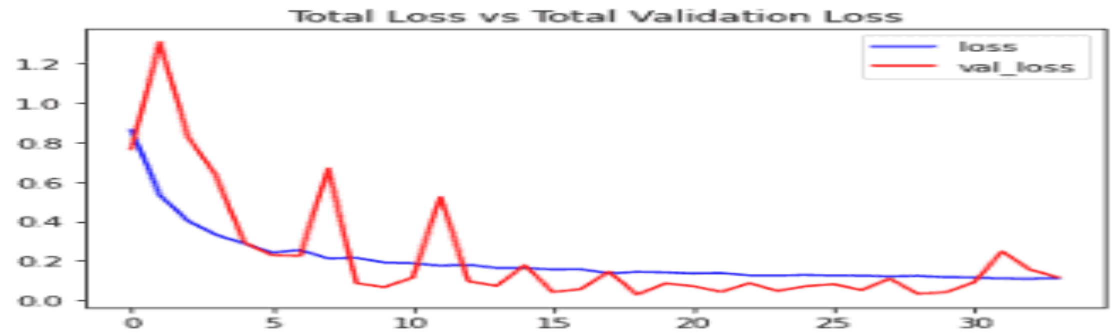
Tabulation:

	Accuracy	Val_accuracy
SIMPLE CNN MODEL	0.9986	0.9977
CONVLSTM MODEL	0.9242	0.8987
LRCN	0.9167	0.9367

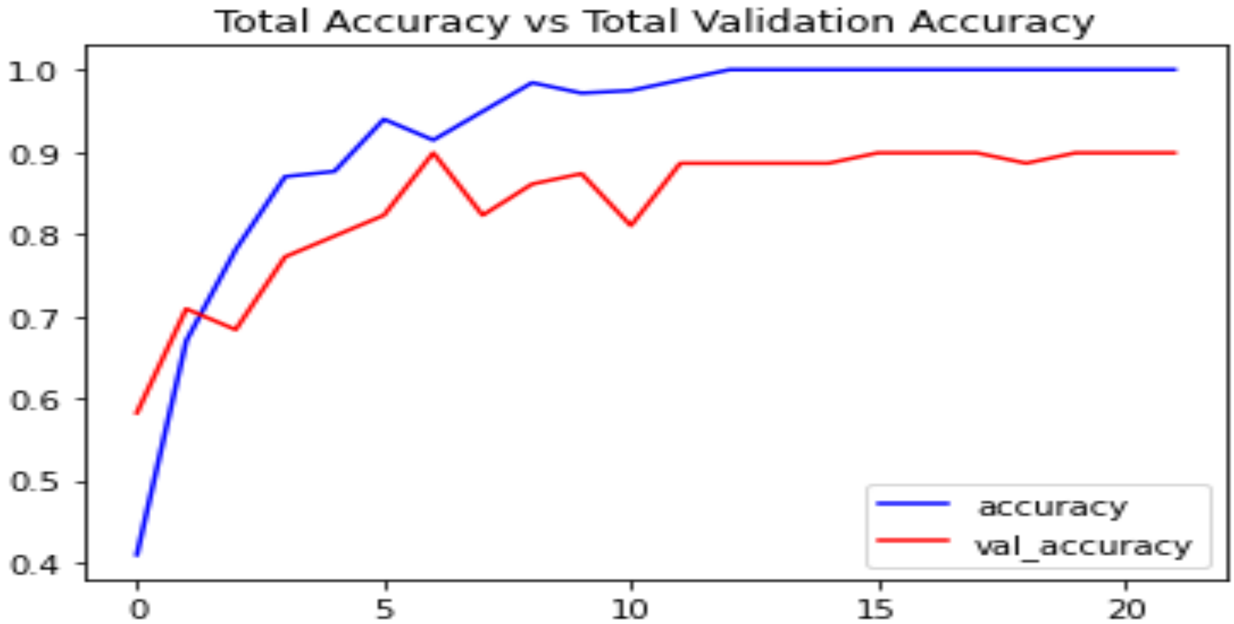
Visualization:

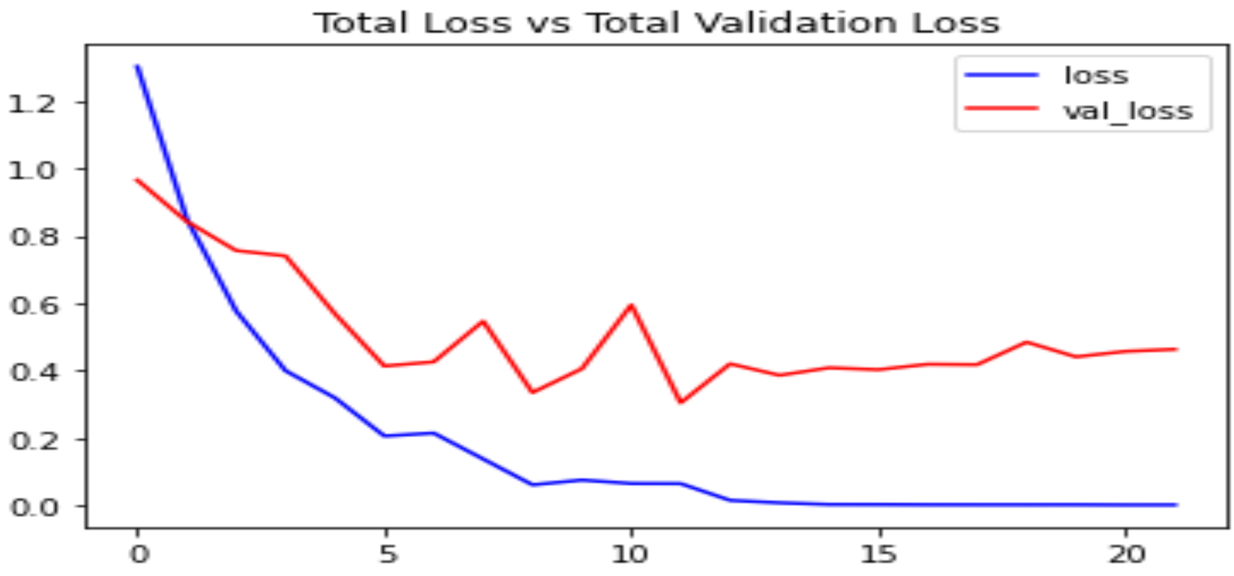
SIMPLE CNN



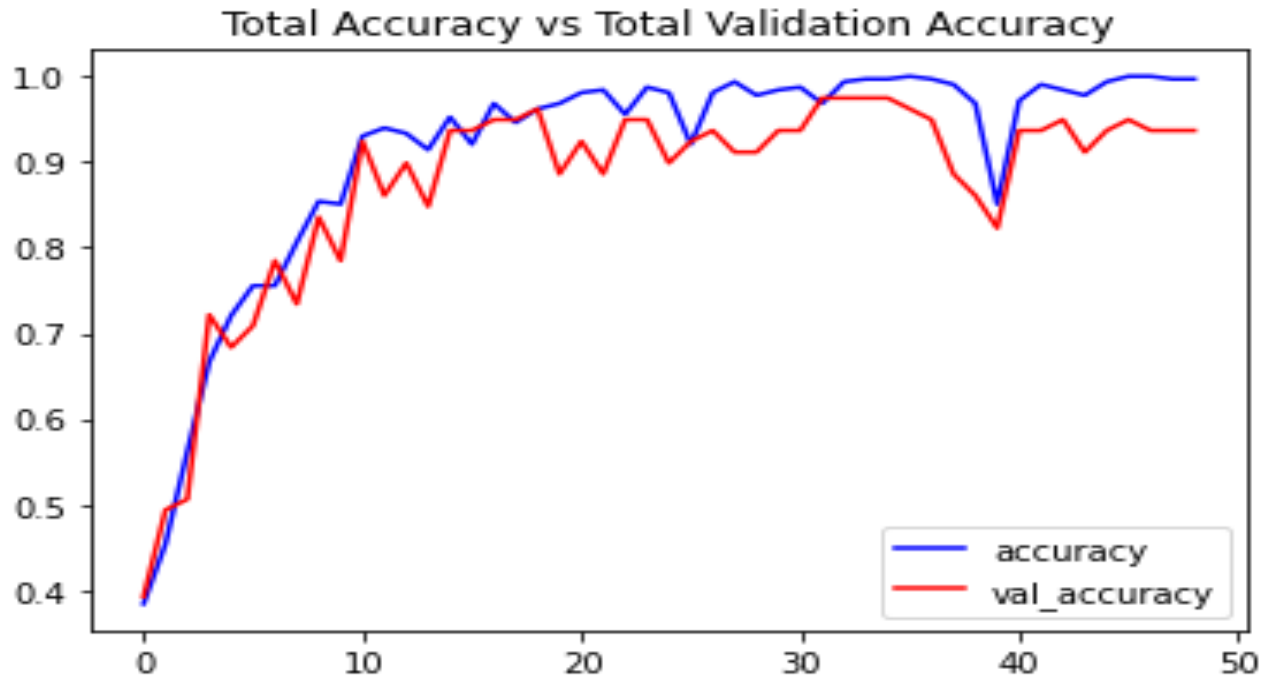

Total Loss vs Total Validation Loss
loss
val_loss
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
5
10
15
20
25
30

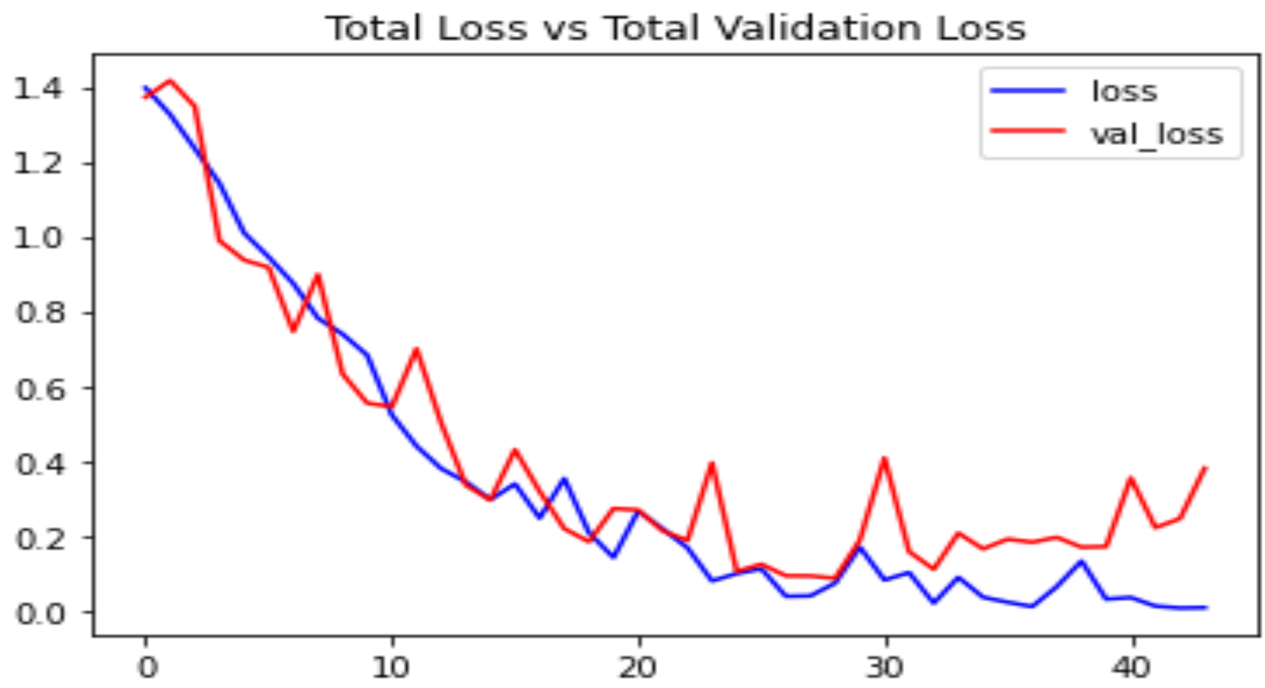
CONVLSTM


Total Accuracy vs Total Validation Accuracy
1.0
0.9
0.8
0.7
0.6
0.5
0.4
accuracy
val_accuracy
0
5
10
15
20


Total Loss vs Total Validation Loss
loss
val_loss
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
5
10
15
20

LRCN


Total Accuracy vs Total Validation Accuracy
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0
10
20
30
40
50
accuracy
val_accuracy


Total Loss vs Total Validation Loss
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
10
20
30
40
loss
val_loss

2) Result analysis, Comment on accuracy, performance

LRCN & SIMPLE CNN – Good Fit Graph

Typical features of the learning curve of a good fit model:

1. Training loss and Validation loss are close to each other with validation loss being slightly greater than the training loss.
2. Initially decreasing training and validation loss and a pretty flat training and validation loss after some point till the end.

CONVLSTM – Overfitting Graph

Typical features of the learning curve of an overfit model:

1. Training loss and Validation loss are far away from each other.
2. Gradually decreasing validation loss (without flattening) upon adding training examples.
3. Very low training loss that's very slightly increasing upon adding training examples.

3) Reasoning about hyperparameters

Hyperparameters used are:
1) number of epochs
2) batch size
3) shuffle
4) validation split
5) callbacks.

1) Number of epochs is the number of times the whole training data is shown to the network while training.
Increase the number of epochs until the validation accuracy starts decreasing even when training accuracy is increasing(overfitting).
In our case for SIMPLE CNN model epochs =50 which gave the good fit model, CONVLSTM model epochs=50 which gave an overfitting model (for this if we would've stopped at around 10th epoch it would have got ridden of overfitting problem.) , for LRCN model epochs=70 which gave a good fit model.

2) Batch Size is the hyperparameter that defines the number of samples to work through before updating the internal model parameters.

In our case we've used a smaller batch size of 4 as to have faster convergence to "good" solutions. Smaller batch sizes allow the model to "start learning before having to see all the data."

3) In our models we used shuffle= True as if you split the data then the resulting sets won't represent the true distribution of the dataset. Therefore, we must shuffle the original dataset in order to minimize the variance and ensure that the model will generalize well to new, unseen data points.

4) In our models we used validation slit= 0.2 , as Validation split helps to improve the model performance by fine-tuning the model after each epoch and it also avoids overfitting.

5) In our models we used callback= early_stopping_callback as it allows you to specify an arbitrary large number of training epochs and stop training once the model performance stops improving on a holdout validation dataset.
It is a form of regularization used to avoid overfitting when training a learner with an iterative method, such as gradient descent. Such methods update the learner to make it better fit the training data with each iteration.

4) Conclusion

Simple CNN- loss: 0.0074 - accuracy: 0.9986

ConvLSTM- loss: 0.3198 - accuracy: 0.9242

LRCN- loss: 0.3220 - accuracy: 0.9167

By comparing the above loss and accuracy of the 3 models we can conclude that simple CNN model performs the best for UCF50 dataset. This is because we've used batch normalization in the Simple CNN model which allows every layer of the network to do learning independently. It also normalizes the output of the previous layers and standardizes the input or outputs.