Gesture Recognition Case study

Team Member:

- Priyanka Patil

Problem Statement

Imagine you are working as a data scientist at a home electronics company which manufactures state of the art smart televisions. You want to develop a cool feature in the smart-TV that can recognise five different gestures performed by the user which will help users control the TV without using a remote.

The gestures are continuously monitored by the webcam mounted on the TV. Each gesture corresponds to a specific command:

Gesture	Corresponding Action
Thumbs Up	Increase the Volume
Thumbs Down	Decrease the Volume
Left swipe	'Jump' backwards 10 seconds.
Right swipe	'Jump' forward 10 seconds
Stop	Pause the movie.

Objectives:

1. **Generator**: The generator should be able to take a batch of videos as input without any error. Steps like cropping, resizing and normalization should be performed successfully.

2. **Model**: Develop a model that can train without any errors which will be judged on the total number of parameters (as the inference(prediction) time should be less) and the accuracy achieved.

3. **Write up**: This should contain the detailed procedure followed in choosing the final model. The write up should start with the reason for choosing the base model, then highlight the reasons and metrics taken into consideration to modify and experiment to arrive at the final model.

MODELS:

Sample model used for initial experiments.

The sample model provides a starting point for the experiments to understand various hyperparameters. Based on the results of the models, we can optimize the hyperparameters and get the better accuracies and reduce the loss.

MODEL – 1

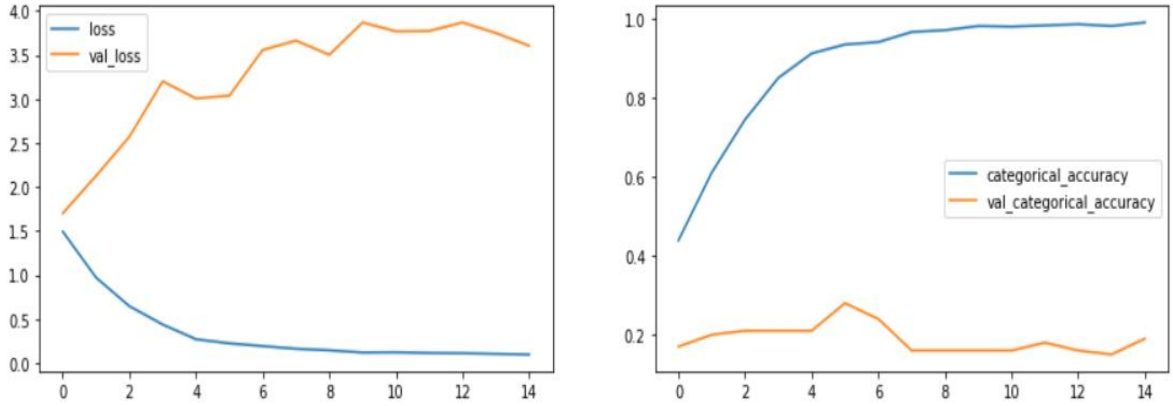
No Data Augmentation:

Batch Size = 40

Epoch = 15

Image Size = 160*160

Training and Validation Accuracy/Loss Plot



Here, we can observe the model is clearly overfitting and there is no improvement in increase of epochs so we now try for CNN LSTM model and see the train and validation accuracies for them so that we can increase or decrease the augmentation accordingly and we try to include the dropout of 25% to address overfitting of model and increase the epochs for better results.

The observations from this model are:

1. Total No. of Parameters = 1,117,061
2. Categorical Accuracy = 99.19%
3. Validation Accuracy = 19.00%

MODEL - 2

CNN LSTM MODEL

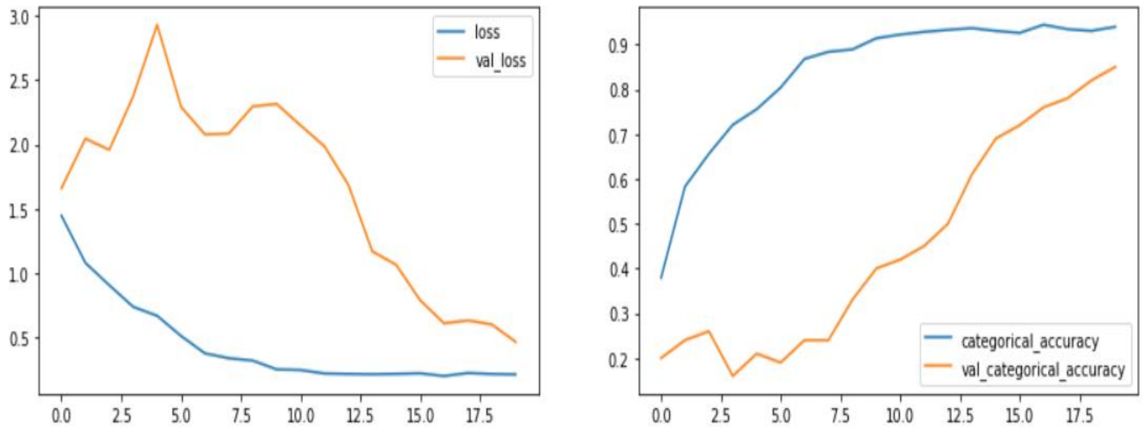
Batch size = 20

Epoch = 20

Image size = 120 * 120 and dropouts = 25%

lstm_cells=128, dense neurons = 128

Training and Validation Accuracy/Loss Plot



Here we achieved a good training accuracy of 93.9% and the validation accuracy of 85.0%.

But there is a lot of difference between training and validation accuracy so let's increase the augmentation and further train the models with different metrices.

The observations from this model are:

1. Total No. of Parameters = 1,657,445
2. Categorical Accuracy = 93.9%
3. Validation Accuracy = 85.00%

MODEL - 3

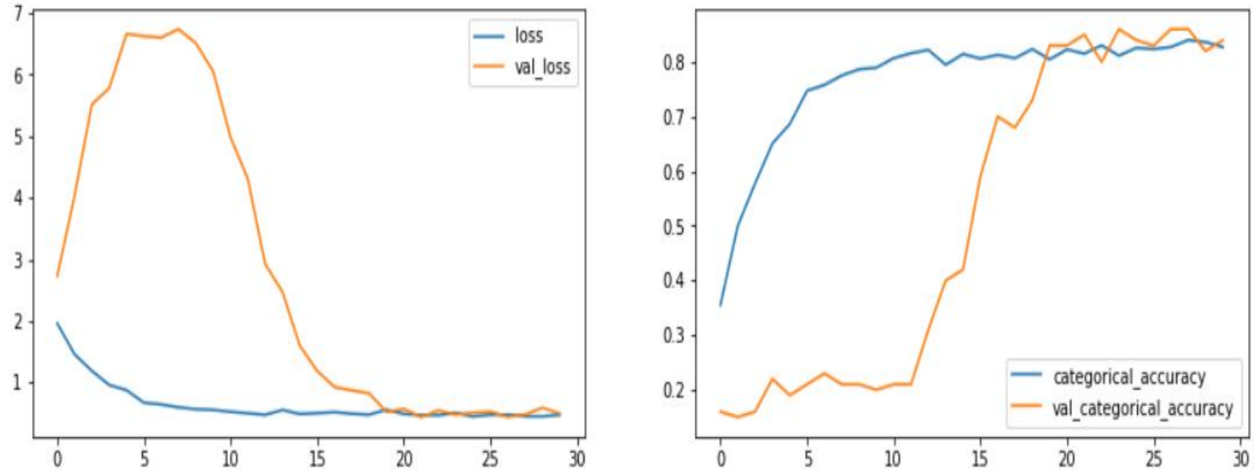
Increased augmentation

Batch size = 20

Epoch = 30

Image size = 160 * 160 and Dropouts = 50%

Training and Validation Accuracy/Loss Plot



Here the model has training accuracy of 82.7% and the validation accuracy of 84.0%.

Here at one instance of epoch both training and validation accuracies got equal and suddenly validation accuracy got increased than train accuracy which is not desirable.

So, let’s try to decrease the image size and increase the batch size and observe the loss and accuracies further.

The observations from this model are:

1. Total No. of Parameters = 3,638,981
2. Categorical Accuracy = 82.7%
3. Validation Accuracy = 84.00%

MODEL - 4

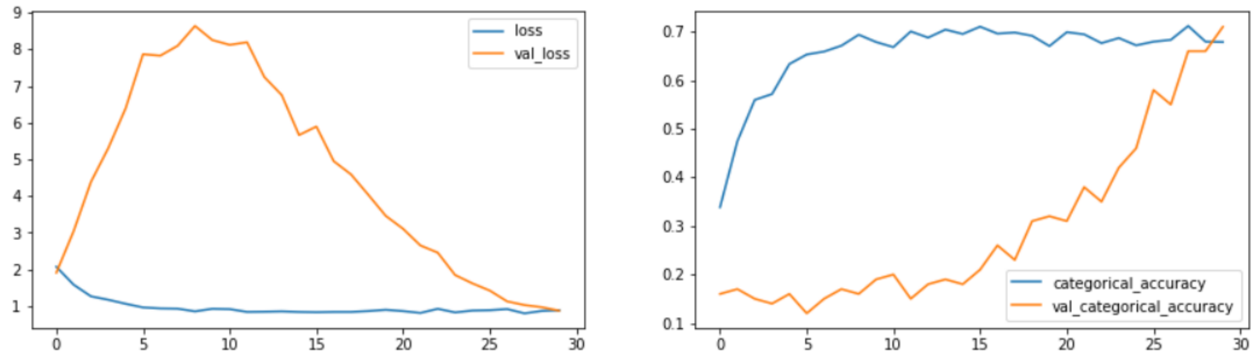
Decreased filter size and image size.

Batch size = 30

Epoch = 30

Image size = 120 * 120 and Dropouts = 50 %

Training and Validation Accuracy/Loss Plot



Here the model has training accuracy of 67.7% and the validation accuracy of 71.0%.¶

Here, we used the filter with (3,3,3) and decreased the image size. We can see both the accuracies are very near but less.

So, let’s try to reduce the number of network parameters and slightly decrease the batch size, epochs. Now observe the loss and accuracies further.

The observations from this model are:

1. Total No. of Parameters = 1,762,613
2. Categorical Accuracy = 67.7%
3. Validation Accuracy = 71.00%

MODEL - 5

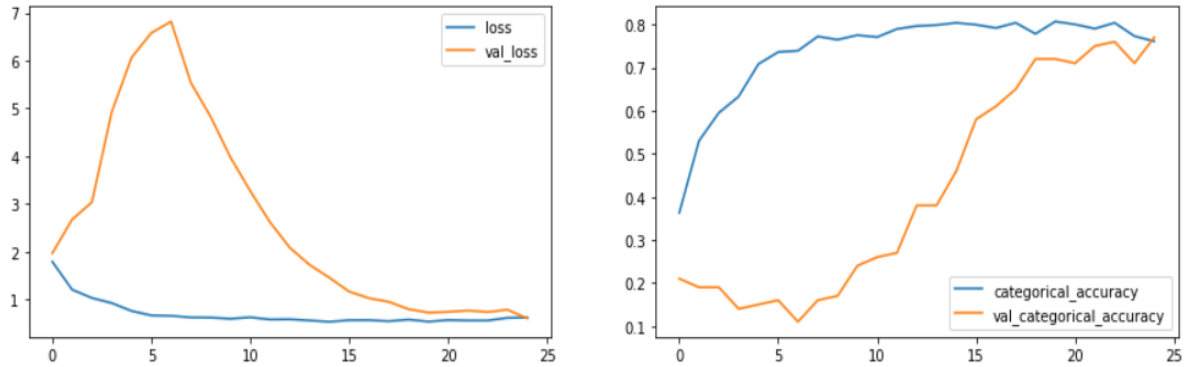
Reducing network parameters

Batch size = 20

Epoch = 25

Image size = 120 * 120 and Dropouts = 25%

Training and Validation Accuracy/Loss Plot



Here the model has training accuracy of 76.62% and the validation accuracy of 76.0%.¶

Here, we used the filter with (3,3,3). We can see both the accuracies are very near and observes pretty good but low in percentages.

So, let’s try CNN LSTM with GRU models.

The observations from this model are:

1. Total No. of Parameters = 504,709
2. Categorical Accuracy = 76.62%
3. Validation Accuracy = 76.00%

MODEL - 6

CNN LSTM with GRU

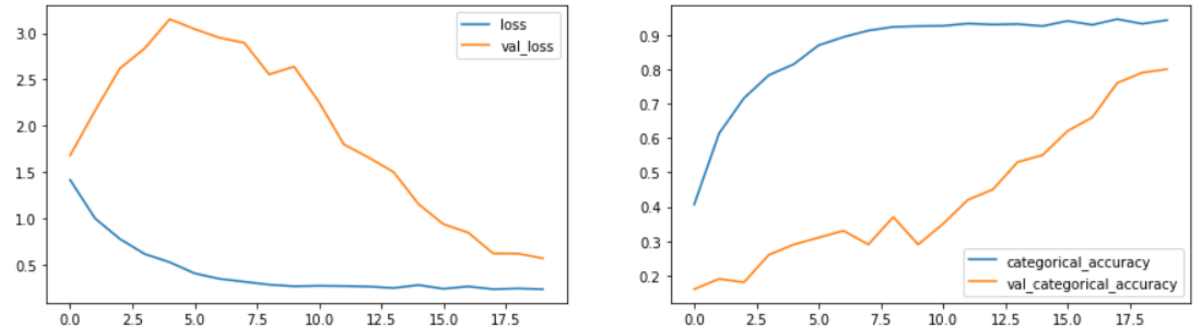
Batch size = 20

Epoch = 20

Image size = 120 * 120 and Dropouts = 25 %

lstm_cells = 128, dense neurons = 128

Training and Validation Accuracy/Loss Plot



Here we achieved a good training accuracy of 93.36% and the validation accuracy of 77.0%.

But there is a lot of difference between training and validation accuracy. As we tried many different combinations, but we are not getting perfect results.

So, let's try Transfer learning which many provide us with best results.

The observations from this model are:

1. Total No. of Parameters = 2,573,925
2. Categorical Accuracy = 93.36%
3. Validation Accuracy = 77.00%

MODEL - 7

Transfer Learning

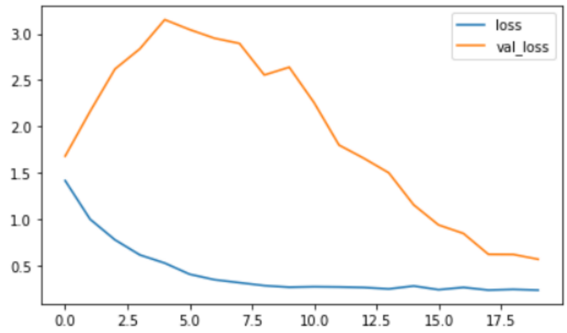
Batch size = 5

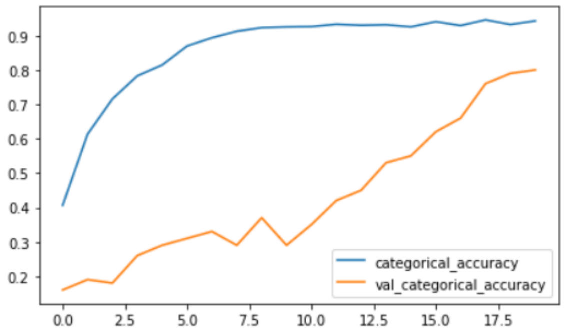
Epoch = 20

Image size = 120 * 120 and Dropouts = 25 %

lstm_cells = 128, dense neurons = 128

Training and Validation Accuracy/Loss Plot





Here we achieved a good training accuracy of 98.19% and the validation accuracy of 78.0%.

- Trainable params: 609,541
- Non-trainable params: 3,230,912

Here, there is a lot of difference between training and validation accuracy. In this model the untrained parameters count is 3,230,912 which very huge and if we try to train all the parameters then we can achieve best results¶

The observations from this model are:

1. Total No. of Parameters = 3,840,453
2. Categorical Accuracy = 98.19%
3. Validation Accuracy = 78.00%

MODEL - 8

Transfer Learning with GRU and training all weights.

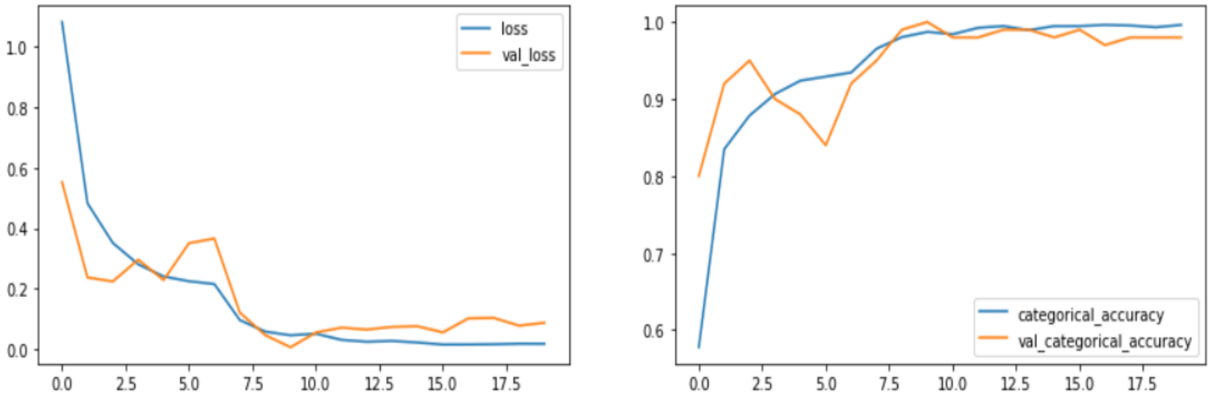
Batch size = 5

Epoch = 20

Image size = 120 * 120 and Dropouts = 25 %

lstm_cells = 128, dense neurons = 128

Training and Validation Accuracy/Loss Plot



Here we achieved a good training accuracy of 99.62% and the validation accuracy of 98.0%.

- Trainable params: 3,669,317
- Non-trainable params: 23,936

Finally, we found the best model!!!

The observations from this model are:

1. Total No. of Parameters = 3,693,253
2. Categorical Accuracy = 99.62%
3. Validation Accuracy = 98.00%

Consolidated Model Summary

Experiment Number	Model	Result	Decision + Explanation
MODEL1	**Conv3D**	Total No. of Parameters = 1,117,061 Categorical Accuracy = 99.19% Validation Accuracy = 19.00%	Highly overfitting. So, Trying with CNN2D and LSTM to further proceed on Augmentation
MODEL2	CNN 2D + LSTM	Total No. of Parameters = 1,657,445 Categorical Accuracy = 93.9% Validation Accuracy = 85.00%	Adding dropout, decreasing Batch size, increasing epochs, Decreasing image size. But There is difference in accuracies So, increasing Augmentation
MODEL3	**Conv3D**	Total No. of Parameters = 3,638,981 Categorical Accuracy = 82.7% Validation Accuracy = 84.00%	Val accuracy is more than Train which Is not reliable. So, decreasing image size, Increasing batch size
MODEL4	CONV 3D	Total No. of Parameters = 1,762,613 Categorical Accuracy = 67.7% Validation Accuracy = 71.00%	Less accuracies, so decreasing batch size, epochs, dropout percentage and use different filter condition
MODEL5	CONV 3D	Total No. of Parameters = 504,709 Categorical Accuracy = 76.62% Validation Accuracy = 76.00%	Best fit model in CONV 3D with Some good accuracies and Less no.of parameters but Let's try some GRU models
MODEL6	CNN 2D + GRU	Total No. of Parameters = 2,573,925 Categorical Accuracy = 93.36% Validation Accuracy = 77.00%	Accuracy is improved but the model is still Overfitting. So, let's try Some transfer learning models

MODEL7	CNN 2D + TL	Total No. of Parameters = 3,840,453 Categorical Accuracy = 98.19% Validation Accuracy = 78.00%	The model is overfitting. So, training weights and try again with transfer learning
MODEL8	CNN 2D + TL	Total No. of Parameters = 3,693,253 Categorical Accuracy = 99.62% Validation Accuracy = 98.00%	Training the model has produced the best model with high accuracies and less loss. Thus a Best fit model

Conclusion

After doing all the experiments, we found that the Model 8 (CNN 2D + TL) With high accuracies.