Following is the highlight of experiments and observations for the Gesture Recognition project.

* Model column consist of snippet of model summary.
* batch normalization play an important role for regularization , however they are used in code but not mentioned explicitly in model column
* We arrived low loss, high and comparable accuracy for training and validation data set with batch size of 10 for all the high performing models and not faced any memory issues.
* We experimented with 16, 18, 20, 30 images per video.
* We experimented with with either 100 by 100, 120 by 120 or 160 by 160 image size.
* Final models are highlighted in green along with their accuracy and loss numbers.
* Data augmentation and parameter reduction seem to help in improving accuracy or loss so we have used in one of the final model.

|  |  |  |  |
| --- | --- | --- | --- |
| **Experiment Number** | **Model** | **Result** | **Decision + Explanation** |
| **1** | **Conv 3D Model with**  **+ 20 frames per video**  **+ 3 by 3 filter size (16, 32, 64, 128 filters conv 3D layers )**  **+ 64 dense layer**  **+ image size 160 by 160**  **+ batch\_size=40**  **+ num\_epochs=20** | **Training Accuracy: 93%**  **Validation Accuracy: 72%** | **Low validation accuracy as compared to training accuracy.**  **Over fitting case.** |
| **2** | **Conv 3D Model with**  **+ 20 frames per video**  **+ 3 by 3 filter size (16, 32, 64, 128 conv 3D layers )**  **+ 256 dense layer**  **+ image size 160 by 160**  **+ batch\_size=20**  **+ num\_epochs=25**  **+ dropout = 0.5** | **Training Accuracy: 60%**  **Validation Accuracy: 54%** | **Low training and validation accuracy** |
| **3** | **Conv 3D Model with**  **+ 30 frames per video**  **+ 2 by 2 filter size (16, 32, 64, 128 conv 3D layers )**  **+ 256 dense layer**  **+ image size 120 by 120**  **+ batch\_size= 30**  **+ num\_epochs=25**  **+ dropout = 0.5** | **Training Accuracy: 60%**  **Validation Accuracy: 20%** | **Low training accuracy**  **Very low validation accuracy** |
| **4** | **Conv 3D Model with**  **+ 16 frames per video**  **+ 3 by 3 filter size (16, 32, 64, 128 conv 3D layers )**  **+ 128 dense layer**  **+ image size 100 by 100**  **+ batch\_size= 20**  **+ num\_epochs=20**  **+ dropout = 0.25** | **Training Accuracy: 58%**  **Validation Accuracy: 32%** | **Low training accuracy**  **Very low validation accuracy**  **Under fitting** |
| **5** | **Conv 3D Model with**  **+ 16 frames per video**  **+ 3 by 3 filter size (16, 32, 64, 128 conv 3D layers )**  **+ 64 dense layer**  **+ image size 120 by 120**  **+ batch\_size= 20**  **+ num\_epochs=25**  **+ dropout = 0.25** | **Training Accuracy: 61%**  **Validation Accuracy: 25%** | **Low training accuracy**  **Very low validation accuracy** |
| **6** | **Conv 3D Model with**  **+ 20 frames per video**  **+ 3 by 3 filter size (16, 32, 64, 128 conv 3D layers )**  **+ 256,128 dense layer**  **+ image size 120 by 120**  **+ batch\_size= 20**  **+ num\_epochs=25**  **+ dropout = 0.25** | **Training Accuracy: 91%**  **Validation Accuracy: 25%** | **High training accuracy**  **Very low validation accuracy**  **Over fitting** |
| **7** | **Time distributed Conv2D with LSTM**  **+ 18 frames per video**  **+ 3 by 3 filter size (16, 32, 64, 128 + + conv 3D layers )**  **+ 64 dense layer**  **+ image size 120 by 120**  **+ batch\_size= 20**  **+ num\_epochs= 20**  **+ dropout = 0.25**  **+ lstm\_cells = 128** | **Training Accuracy: 66%**  **Validation Accuracy: 31%** | **Avg training accuracy**  **Low validation accuracy**  **Marginal underfitting** |
| **8** | **Transfer Learning - MobileNet Conv2d + GRU**  **+ 20 frames per video**  **+ 128 dense layer**  **+ image size 120 by 120**  **+ batch\_size= 10**  **+ num\_epochs= 20**  **+ dropout = 0.25**  **+ gru\_cells = 128**  **+ gru = true** | **Training Loss: 0.0190**  **Training Accuracy: 99.40%**  **Validation Loss: 0.0832**  **Validation Accuracy: 97%** | **Total params: 3,693,253**  **Low loss ,high and comparable accuracy on both training and validation data sets.** |
| **9** | **MobileNet Conv2d + GRU with data augmentation**  **+ 30 frames per video**  **+ 128 dense layer**  **+ image size 120 by 120**  **+ batch\_size= 10**  **+ num\_epochs= 20**  **+ dropout = 0.25**  **+ gru\_cells = 128**  **+ gru = true**  **+ data augmentation = true** | **Training Loss: 0.0125**  **Training Accuracy: 99.85%**  **Validation Loss: 0.1149**  **Validation Accuracy: 98%** | **Total params: 3,693,253**  **With augmentation, we were able to increase the validation accuracy by 1 %.**  **Low loss ,high and comparable accuracy on both training and validation data sets.** |
| **10** | **MobileNet Conv2d + GRU with data augmentation + parameter reduction**  **+ 20 frames per video**  **+ 64 dense layer**  **+ image size 100 by 100**  **+ batch\_size= 10**  **+ num\_epochs= 20**  **+ dropout = 0.25**  **+ gru\_cells = 64**  **+ gru = true**  **+ data augmentation = true**  **+ parameter reduction = yes** | **Training Loss: 0.0591**  **Training Accuracy: 97.74%**  **Validation Loss: 0.0478**  **Validation Accuracy: 99%** | **Total params: 3,446,725**  **Though able to reduce the number of parameter by 246,528. But more work can be done wrt parameter reduction**  **Low loss ,higher and greater accuracy of training data set wrt validation data sets.** |
| **11** | **Another iteration of Model 10 has been performed with added parameter reduction.** | **Training Loss: 0.1706**  **Training Accuracy: 96.23%**  **Validation Loss: 0.1213**  **Validation Accuracy: 98%** | **Avg model performance. Though we can see low losses and high accuracy but the plot shows does not show smooth progression.** |
| **12** | **MobileNet Conv2d + LSTM without data augmentation** **+ 30 frames per video**  **+ 128 dense layer**  **+ image size 120 by 120**  **+ batch\_size= 10**  **+ num\_epochs= 20**  **+ dropout = 0.25**  **+ lstm\_cells = 128**  **+ lstm = true**  **+ data augmentation = false** | **Training Loss: 0.2160**  **Training Accuracy: 92.61%**  **Validation Loss: 0.2714**  **Validation Accuracy: 88%** | **Total params: 3,840,453**  **Overall a good model but not performed as Model 9 and 10 above.** |