Gesture Recognition results below explain the different experiments conducted in Conv3D and Conv2D+LSTM networks to classify Hand recognition out of the given videos image frames from each 663 videos into 5 different classes/categories.

**Below are the experiments for CONV3D model.**

**Model code is in zipped folder.**

1. Model with below parameters.

1. Batch\_size = 50
2. No of epocs =7
3. No of frames = 30
4. Input Image size = 100X100
5. LR = 0.01
6. Optimizer = Adam
7. 3 CONV3D layers with batch normalization and dropout.
8. Total params: 7,162,981
9. Trainable params: 7,162,277
10. Non-trainable params: 704

Getting ResourceExhaustedError while training model

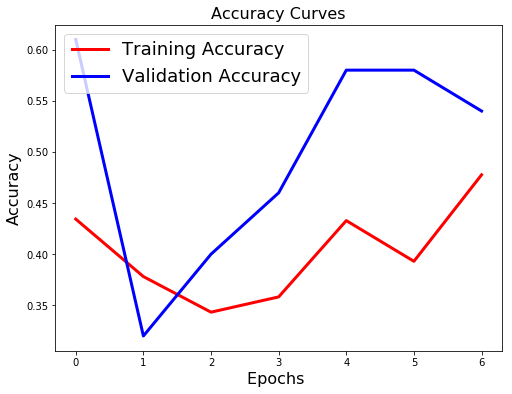
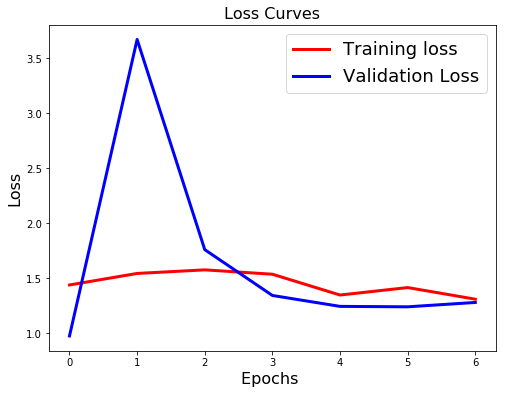
Explanation: Since number of trainable parameters are large, getting out of memory error from nimblebox.

2. Decreasing Batch size to 10: Training Accuracy = 48% and Validation Accuracy = 54%

1. Batch\_size = 10
2. No of epocs =7
3. No of frames = 10
4. Input Image size = 100X100
5. LR = 0.01
6. Optimizer = Adam
7. 3 CONV3D layers with batch normalization and dropout.
8. Total params: 2,444,389
9. Trainable params: 2,443,685
10. Non-trainable params: 704

Epoch 7/7

67/67 [==============================] - 8s 118ms/step - loss: 1.3075 - categorical\_accuracy: 0.4776 - val\_loss: 1.2779 - val\_categorical\_accuracy: 0.5400



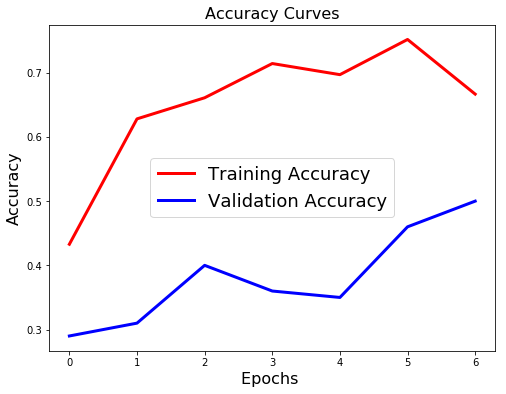
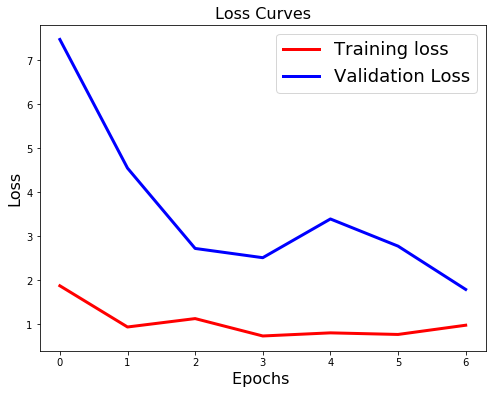
Explaination: Training accuracy is low and need to tweek the parameters to get the better model.

3. Increasing Batch size to 100: Training Accuracy = 68% and Validation Accuracy = 50%

1. Batch\_size = 100
2. No of epocs =7
3. No of frames = 10
4. Input Image size = 100X100
5. LR = 0.01
6. Optimizer = Adam
7. 3 CONV3D layers with batch normalization and dropout.
8. Total params: 2,444,389
9. Trainable params: 2,443,685
10. Non-trainable params: 704

Epoch 7/7

7/7 [==============================] - 1s 139ms/step - loss: 0.9762 - categorical\_accuracy: **0.6667** - val\_loss: 1.7890 - val\_categorical\_accuracy: **0.5000**



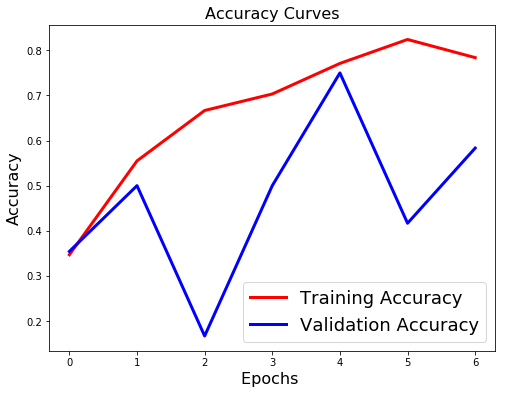
Explaination: Validation accuracy is low, and the model is getting overfitted.

4. Changing optimizer from Adam to SGD: Training accuracy = 78% and validation accuracy = 58%

1. Batch\_size = 48
2. No of epocs =7
3. No of frames = 10
4. Input Image size = 100X100
5. LR = 0.01
6. Optimizer = SGD
7. 3 CONV3D layers with batch normalization and dropout.
8. Total params: 2,444,389
9. Trainable params: 2,443,685
10. Non-trainable params: 704

Epoch 7/7

14/14 [==============================] - 17s 1s/step - loss: 0.5457 - categorical\_accuracy: 0.7839 - val\_loss: 1.0913 - val\_categorical\_accuracy: 0.5833



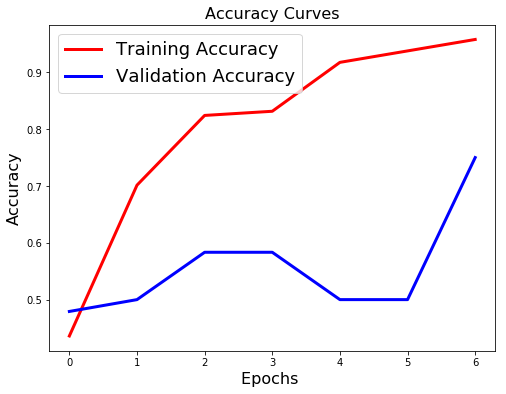
Explaination: Validation accuracy is low, and the model is getting overfitted.

5. Reducing learning rate from 0.01 to 0.001: Training accuracy = 96% and validation accuracy = 75%

1. Batch\_size = 48
2. No of epocs =7
3. No of frames = 10
4. Input Image size = 100X100
5. LR = 0.001
6. Optimizer = Adam
7. 3 CONV3D layers with batch normalization and dropout.
8. Total params: 2,444,389
9. Trainable params: 2,443,685
10. Non-trainable params: 704

Epoch 7/7

14/14 [==============================] - 17s 1s/step - loss: 0.1823 - categorical\_accuracy: 0.9579 - val\_loss: 1.0731 - val\_categorical\_accuracy: 0.7500



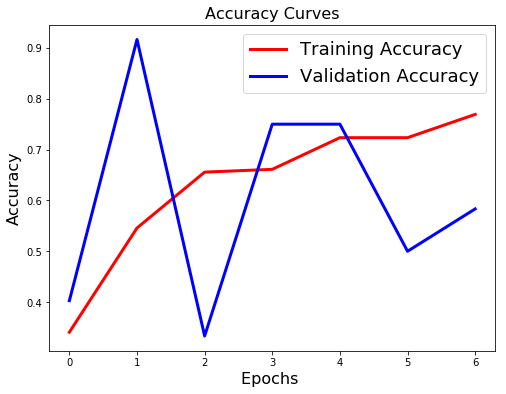
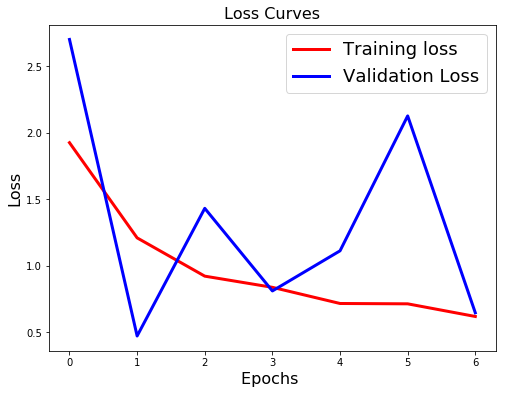
Explaination: Validation accuracy is low, and the model is getting overfitted.

6. Dropout value = 0.50 in all layers: Training accuracy = 77% and validation accuracy = 58%

1. Batch\_size = 48
2. No of epocs =7
3. No of frames = 10
4. Input Image size = 100X100
5. LR = 0.01
6. Optimizer = Adam
7. 3 CONV3D layers with batch normalization and dropout.
8. Total params: 2,444,389
9. Trainable params: 2,443,685
10. Non-trainable params: 704

Epoch 7/7

14/14 [==============================] - 16s 1s/step - loss: 0.6167 - categorical\_accuracy: 0.7692 - val\_loss: 0.6444 - val\_categorical\_accuracy: 0.5833

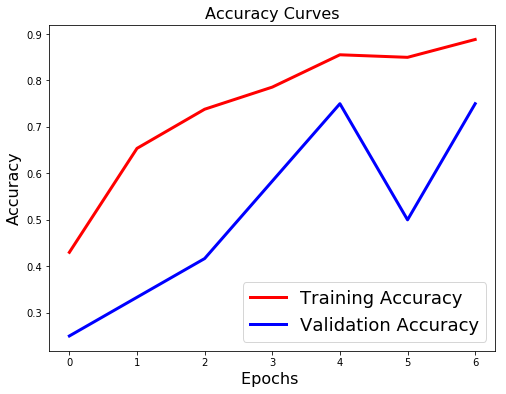
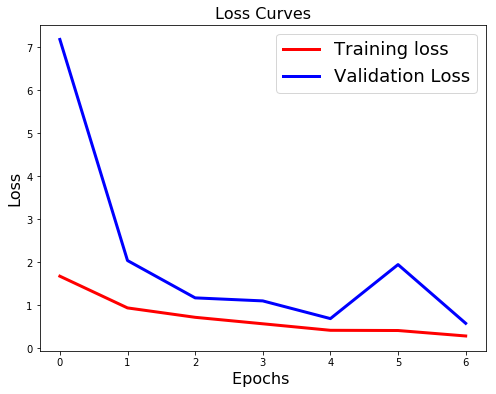


Explaination: Validation accuracy is low, and the model is getting overfitted.

7. Decreasing the dropout value from 0.50 to 0.25 in all layers: Training accuracy = 89% and validation accuracy = 75%

1. Batch\_size = 48
2. No of epocs =7
3. No of frames = 10
4. Input Image size = 100X100
5. LR = 0.01
6. Optimizer = Adam
7. 3 CONV3D layers with batch normalization and dropout.
8. Total params: 2,444,389
9. Trainable params: 2,443,685
10. Non-trainable params: 704

14/14 [==============================] - 17s 1s/step - loss: 0.2900 - categorical\_accuracy: 0.8883 - val\_loss: 0.5842 - val\_categorical\_accuracy: 0.7500



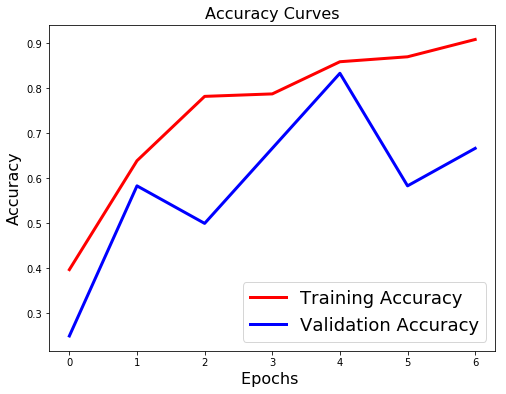
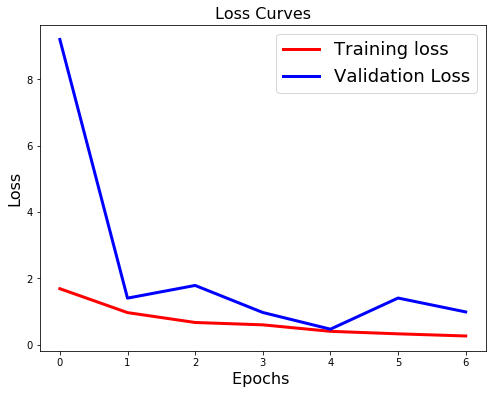
Explaination: Good model but we are still experimenting to get the best model. This is slightly overfitted.

8. Changing the input shape in generator to feed (no. of frames, image length, image width, channels) to (image length, image width, no. of frames, channels): Training accuracy = 91% and validation accuracy = 61%

1. Batch\_size = 48
2. No of epocs =7
3. No of frames = 10
4. Input Image size = 100X100
5. LR = 0.01
6. Optimizer = Adam
7. 3 CONV3D layers with batch normalization and dropout.
8. Total params: 2,444,389
9. Trainable params: 2,443,685
10. Non-trainable params: 704

Epoch 7/7

14/14 [==============================] - 17s 1s/step - loss: 0.2591 - categorical\_accuracy: 0.9084 - val\_loss: 0.9842 - val\_categorical\_accuracy: 0.6667



Explaination: Validation accuracy is low, highly overfitted model.

9. Increasing the no. of epocs from 7 to 10: Training accuracy =97% and validation accuracy = 83%

1. Batch\_size = 48
2. No of epocs =10
3. No of frames = 10
4. Input Image size = 100X100
5. LR = 0.01
6. Optimizer = Adam
7. 3 CONV3D layers with batch normalization and dropout.
8. Total params: 2,444,389
9. Trainable params: 2,443,685
10. Non-trainable params: 704

Epoch 7/10

14/14 [==============================] - 17s 1s/step - loss: 0.2466 - categorical\_accuracy: 0.9194 - val\_loss: 0.8600 - val\_categorical\_accuracy: 0.7500

Epoch 00007: val\_loss did not improve from 0.44703

Epoch 8/10

14/14 [==============================] - 17s 1s/step - loss: 0.1669 - categorical\_accuracy: 0.9542 - val\_loss: 0.8410 - val\_categorical\_accuracy: 0.7500

Epoch 00008: val\_loss did not improve from 0.44703

Epoch 00008: ReduceLROnPlateau reducing learning rate to 0.0009999999776482583.

Epoch 9/10

14/14 [==============================] - 16s 1s/step - loss: 0.1218 - categorical\_accuracy: 0.9615 - val\_loss: 1.7522 - val\_categorical\_accuracy: 0.5833

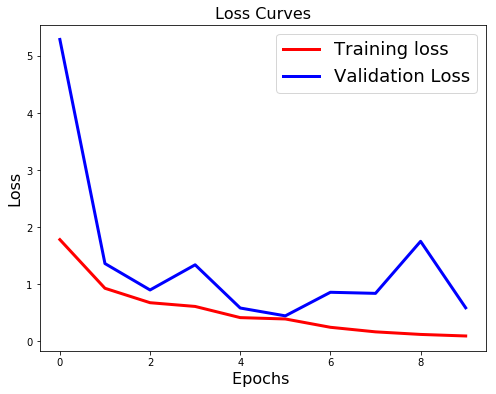
Epoch 00009: val\_loss did not improve from 0.44703

Epoch 10/10

14/14 [==============================] - 17s 1s/step - loss: 0.0947 - categorical\_accuracy: 0.9780 - val\_loss: 0.5880 - val\_categorical\_accuracy: 0.8333

Epoch 00010: val\_loss did not improve from 0.44703

Epoch 00010: ReduceLROnPlateau reducing learning rate to 0.0001.



Explaination: After 7th epoch though the training accuracy increases the validation accuracy decreases and then increase drastically.

10. Removing batch normalization in all the layers in the model architecture: Training accuracy =20% and validation accuracy = 17%

1. Batch\_size = 48
2. No of epocs =10
3. No of frames = 10
4. Input Image size = 100X100
5. LR = 0.01
6. Optimizer = Adam
7. 3 CONV3D layers with batch normalization and dropout.
8. Total params: 2,442,981
9. Trainable params: 2,442,981
10. Non-trainable params: 0

Epoch 10/10

14/14 [==============================] - 17s 1s/step - loss: 12.9299 - categorical\_accuracy: 0.1978 - val\_loss: 13.4317 - val\_categorical\_accuracy: 0.1667



Explaination: Poor performing model.

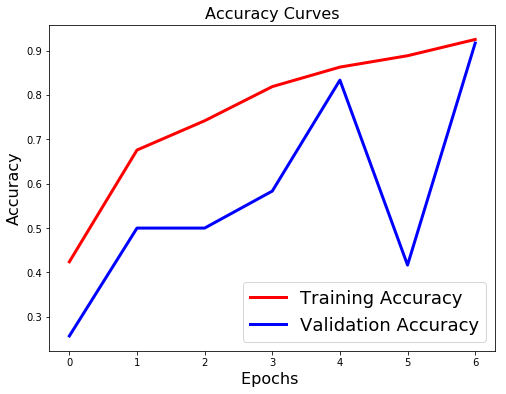
11. Best model: Training accuracy = 92% and validation accuracy = 92%

1. Batch\_size = 48
2. No of epocs =7
3. No of frames = 10
4. Input Image size = 100X100
5. LR = 0.01
6. Optimizer = Adam
7. 3 CONV3D layers with batch normalization and dropout.
8. Total params: 2,444,389
9. Trainable params: 2,443,685
10. Non-trainable params: 704

Epoch 7/7

14/14 [==============================] - 16s 1s/step - loss: 0.2067 - categorical\_accuracy: 0.9249 - val\_loss: 0.2138 - val\_categorical\_accuracy: 0.9167

Epoch 00007: val\_loss improved from 0.47606 to 0.21384, saving model to model\_init\_2020-05-0411\_59\_51.570587/model-00007-0.20671-0.92491-0.21384-0.91667.h5



Explaination: This model is pretty good with high training and validation accuracies.

**Below are the experiments for CONV2D + LSTM model.**

**Github link for model code:**

<https://github.com/pjain2411/Gesture-Recognition-using-CONV2D-and-LSTM/blob/master/Neural_Nets_Project_CONV2D_LSTM.ipynb>

1. Model with below parameters.

1. Batch\_size = 50
2. No of epocs = 10
3. No of frames = 10
4. Input Image size = 100X100
5. LR = 0.0001
6. Optimizer = Adam
7. 3 CONV2D layers and LSTM with batch normalization and dropout.
8. Total params: 4,778,021
9. Trainable params: 4,777,573
10. Non-trainable params: 448

Epoch 10/10

14/14 [==============================] - 22s 2s/step - loss: 0.6683 - categorical\_accuracy: 0.8100 - val\_loss: 1.0388 - val\_categorical\_accuracy: 0.6100

Explanation: Though the training accuracy is pretty good the model is slightly overfitted.

\*---------------------------END-------------------------\*