Gesture Recognition

Problem Statement

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

Solution

Before running any models, generator function was prepared which was loading the data into memory in batches for processing and performing some preprocessing steps like:

1. Image Cropping (to 120x120 if the size is 120x160)
2. Image Resizing (to 120x120 if the size is 360x360)
3. Image Normalization (all color values should be in the range 0 and 1)

Following models were prepared on the final training data:

Exec #	Model	Result	Decision + Explanation	
1	**Conv3D**	Accuracy: 18.5 Val Accuracy: 17.0	Explanation - Model is not learning Decision - Increase the number of parameters	**Model: "sequential"** **Layer (type)** / **Output Shape** / **Param #** conv3d (Conv3D) — (None, 23, 118, 118, 32) — 2,624 max_pooling3d (MaxPooling3D) — (None, 11, 59, 59, 32) — 0 dropout (Dropout) — (None, 11, 59, 59, 32) — 0 flatten (Flatten) — (None, 1225312) — 0 dense (Dense) — (None, 32) — 39,210,016 dropout_1 (Dropout) — (None, 32) — 0 dense_1 (Dense) — (None, 5) — 165 **Total params:** 39,212,805 (149.58 MB) **Trainable params:** 39,212,805 (149.58 MB) **Non-trainable params:** 0 (0.00 B)
2	**Conv3D**	Accuracy: 68.8 Val Accuracy: 37.5	Explanation - Model has started learning but has started overfitting as well Decision - Increase the number of Conv3d layers	**Model: "sequential_1"** **Layer (type)** / **Output Shape** / **Param #** conv3d_1 (Conv3D) — (None, 23, 118, 118, 32) — 2,624 max_pooling3d_1 (MaxPooling3D) — (None, 11, 59, 59, 32) — 0 dropout_2 (Dropout) — (None, 11, 59, 59, 32) — 0 flatten_1 (Flatten) — (None, 1225312) — 0 dense_2 (Dense) — (None, 32) — 39,210,016 dropout_3 (Dropout) — (None, 32) — 0 dense_3 (Dense) — (None, 256) — 8,448 dropout_4 (Dropout) — (None, 256) — 0 dense_4 (Dense) — (None, 5) — 1,285 **Total params:** 39,222,373 (149.62 MB) **Trainable params:** 39,222,373 (149.62 MB) **Non-trainable params:** 0 (0.00 B)

<table>
<tr>
<td>3</td>
<td>Conv3D</td>
<td>Accuracy: 21.3
Val Accuracy: 27.7</td>
<td>Explanation - Model has stopped learning, while overcoming overfitting

Decision - Add normalization after every layer</td>
<td>
Model: "sequential_2"
<table>
<tr><th>Layer (type)</th><th>Output Shape</th><th>Param #</th></tr>
<tr><td>conv3d_2 (Conv3D)</td><td>(None, 23, 118, 118, 32)</td><td>2,624</td></tr>
<tr><td>max_pooling3d_2 (MaxPooling3D)</td><td>(None, 11, 59, 59, 32)</td><td>0</td></tr>
<tr><td>dropout_5 (Dropout)</td><td>(None, 11, 59, 59, 32)</td><td>0</td></tr>
<tr><td>conv3d_3 (Conv3D)</td><td>(None, 9, 57, 57, 64)</td><td>55,360</td></tr>
<tr><td>max_pooling3d_3 (MaxPooling3D)</td><td>(None, 4, 28, 28, 64)</td><td>0</td></tr>
<tr><td>dropout_6 (Dropout)</td><td>(None, 4, 28, 28, 64)</td><td>0</td></tr>
<tr><td>flatten_2 (Flatten)</td><td>(None, 200704)</td><td>0</td></tr>
<tr><td>dense_5 (Dense)</td><td>(None, 32)</td><td>6,422,560</td></tr>
<tr><td>dropout_7 (Dropout)</td><td>(None, 32)</td><td>0</td></tr>
<tr><td>dense_6 (Dense)</td><td>(None, 256)</td><td>8,448</td></tr>
<tr><td>dropout_8 (Dropout)</td><td>(None, 256)</td><td>0</td></tr>
<tr><td>dense_7 (Dense)</td><td>(None, 5)</td><td>1,285</td></tr>
</table>
Total params: 6,490,277 (24.76 MB)
Trainable params: 6,490,277 (24.76 MB)
Non-trainable params: 0 (0.00 B)
</td>
</tr>
<tr>
<td>4</td>
<td>Conv3D</td>
<td>Accuracy: 94.0
Val Accuracy: 63.4</td>
<td>Explanation - Model is learning again but with overfitting

Decision - Increase the number of Conv3d layers</td>
<td>
Model: "sequential_3"
<table>
<tr><th>Layer (type)</th><th>Output Shape</th><th>Param #</th></tr>
<tr><td>conv3d_4 (Conv3D)</td><td>(None, 23, 118, 118, 32)</td><td>2,624</td></tr>
<tr><td>max_pooling3d_4 (MaxPooling3D)</td><td>(None, 11, 59, 59, 32)</td><td>0</td></tr>
<tr><td>batch_normalization (BatchNormalization)</td><td>(None, 11, 59, 59, 32)</td><td>128</td></tr>
<tr><td>dropout_9 (Dropout)</td><td>(None, 11, 59, 59, 32)</td><td>0</td></tr>
<tr><td>conv3d_5 (Conv3D)</td><td>(None, 9, 57, 57, 64)</td><td>55,360</td></tr>
<tr><td>max_pooling3d_5 (MaxPooling3D)</td><td>(None, 4, 28, 28, 64)</td><td>0</td></tr>
<tr><td>batch_normalization_1 (BatchNormalization)</td><td>(None, 4, 28, 28, 64)</td><td>256</td></tr>
<tr><td>dropout_10 (Dropout)</td><td>(None, 4, 28, 28, 64)</td><td>0</td></tr>
<tr><td>flatten_3 (Flatten)</td><td>(None, 200704)</td><td>0</td></tr>
<tr><td>dense_8 (Dense)</td><td>(None, 32)</td><td>6,422,560</td></tr>
<tr><td>dropout_11 (Dropout)</td><td>(None, 32)</td><td>0</td></tr>
<tr><td>batch_normalization_2 (BatchNormalization)</td><td>(None, 32)</td><td>128</td></tr>
<tr><td>dense_9 (Dense)</td><td>(None, 256)</td><td>8,448</td></tr>
<tr><td>dropout_12 (Dropout)</td><td>(None, 256)</td><td>0</td></tr>
<tr><td>batch_normalization_3 (BatchNormalization)</td><td>(None, 256)</td><td>1,024</td></tr>
<tr><td>dense_10 (Dense)</td><td>(None, 5)</td><td>1,285</td></tr>
</table>
Total params: 6,491,813 (24.76 MB)
Trainable params: 6,491,045 (24.76 MB)
Non-trainable params: 768 (3.00 KB)
</td>
</tr>
<tr>
<td>5</td>
<td>Conv3D</td>
<td>Accuracy: 83.9
Val Accuracy: 46.4</td>
<td>Explanation - Model is still overfitting

Decision - Change order of Conv3d layers along with normalization</td>
<td>
Model: "sequential_5"
<table>
<tr><th>Layer (type)</th><th>Output Shape</th><th>Param #</th></tr>
<tr><td>conv3d_12 (Conv3D)</td><td>(None, 25, 120, 120, 32)</td><td>2,624</td></tr>
<tr><td>conv3d_13 (Conv3D)</td><td>(None, 25, 120, 120, 32)</td><td>27,680</td></tr>
<tr><td>max_pooling3d_9 (MaxPooling3D)</td><td>(None, 12, 60, 60, 32)</td><td>0</td></tr>
<tr><td>batch_normalization_9 (BatchNormalization)</td><td>(None, 12, 60, 60, 32)</td><td>128</td></tr>
<tr><td>dropout_18 (Dropout)</td><td>(None, 12, 60, 60, 32)</td><td>0</td></tr>
<tr><td>conv3d_14 (Conv3D)</td><td>(None, 12, 60, 60, 64)</td><td>55,360</td></tr>
<tr><td>conv3d_15 (Conv3D)</td><td>(None, 12, 60, 60, 64)</td><td>110,656</td></tr>
<tr><td>max_pooling3d_10 (MaxPooling3D)</td><td>(None, 6, 30, 30, 64)</td><td>0</td></tr>
<tr><td>batch_normalization_10 (BatchNormalization)</td><td>(None, 6, 30, 30, 64)</td><td>256</td></tr>
<tr><td>dropout_19 (Dropout)</td><td>(None, 6, 30, 30, 64)</td><td>0</td></tr>
<tr><td>conv3d_16 (Conv3D)</td><td>(None, 6, 30, 30, 128)</td><td>221,312</td></tr>
<tr><td>conv3d_17 (Conv3D)</td><td>(None, 6, 30, 30, 128)</td><td>442,496</td></tr>
<tr><td>max_pooling3d_11 (MaxPooling3D)</td><td>(None, 3, 15, 15, 128)</td><td>0</td></tr>
<tr><td>batch_normalization_11 (BatchNormalization)</td><td>(None, 3, 15, 15, 128)</td><td>512</td></tr>
<tr><td>dropout_20 (Dropout)</td><td>(None, 3, 15, 15, 128)</td><td>0</td></tr>
<tr><td>flatten_5 (Flatten)</td><td>(None, 86400)</td><td>0</td></tr>
<tr><td>dense_14 (Dense)</td><td>(None, 32)</td><td>2,764,832</td></tr>
<tr><td>dropout_21 (Dropout)</td><td>(None, 32)</td><td>0</td></tr>
<tr><td>batch_normalization_12 (BatchNormalization)</td><td>(None, 32)</td><td>128</td></tr>
<tr><td>dense_15 (Dense)</td><td>(None, 256)</td><td>8,448</td></tr>
<tr><td>dropout_22 (Dropout)</td><td>(None, 256)</td><td>0</td></tr>
<tr><td>batch_normalization_13 (BatchNormalization)</td><td>(None, 256)</td><td>1,024</td></tr>
<tr><td>dense_16 (Dense)</td><td>(None, 5)</td><td>1,285</td></tr>
</table>
Total params: 3,636,741 (13.87 MB)
Trainable params: 3,635,717 (13.87 MB)
Non-trainable params: 1,024 (4.00 KB)
</td>
</tr>
</table>

6	**Conv3D**	Accuracy: 72.8 Val Accuracy: 56.3	Explanation - Model overfitting has reduced Decision - Replace flatten with global average pooling	**Model: "sequential_6"** Layer (type) / Output Shape / Param # conv3d_18 (Conv3D) — (None, 25, 120, 120, 16) — 1,312 batch_normalization_14 (BatchNormalization) — (None, 25, 120, 120, 16) — 64 conv3d_19 (Conv3D) — (None, 25, 120, 120, 16) — 6,928 batch_normalization_15 (BatchNormalization) — (None, 25, 120, 120, 16) — 64 max_pooling3d_12 (MaxPooling3D) — (None, 12, 60, 60, 16) — 0 conv3d_20 (Conv3D) — (None, 12, 60, 60, 32) — 13,856 batch_normalization_16 (BatchNormalization) — (None, 12, 60, 60, 32) — 128 conv3d_21 (Conv3D) — (None, 12, 60, 60, 32) — 27,680 batch_normalization_17 (BatchNormalization) — (None, 12, 60, 60, 32) — 128 max_pooling3d_13 (MaxPooling3D) — (None, 6, 30, 30, 32) — 0 conv3d_22 (Conv3D) — (None, 6, 30, 30, 64) — 55,360 batch_normalization_18 (BatchNormalization) — (None, 6, 30, 30, 64) — 256 conv3d_23 (Conv3D) — (None, 6, 30, 30, 64) — 110,656 batch_normalization_19 (BatchNormalization) — (None, 6, 30, 30, 64) — 256 max_pooling3d_14 (MaxPooling3D) — (None, 6, 15, 15, 64) — 0 conv3d_24 (Conv3D) — (None, 4, 13, 13, 128) — 221,312 conv3d_25 (Conv3D) — (None, 2, 11, 11, 128) — 442,496 max_pooling3d_15 (MaxPooling3D) — (None, 2, 5, 5, 128) — 0 flatten_6 (Flatten) — (None, 6400) — 0 dense_17 (Dense) — (None, 128) — 819,328 batch_normalization_20 (BatchNormalization) — (None, 128) — 512 dropout_23 (Dropout) — (None, 128) — 0 dense_18 (Dense) — (None, 512) — 66,048 batch_normalization_21 (BatchNormalization) — (None, 512) — 2,048 dense_19 (Dense) — (None, 5) — 2,565 **Total params:** 1,770,997 (6.76 MB) **Trainable params:** 1,769,269 (6.75 MB) **Non-trainable params:** 1,728 (6.75 KB)
7	**Conv3D**	Accuracy: 54.3 Val Accuracy: 55.4	Explanation - Model is not learning much but overfitting is not there Decision - Use time distributed Conv2D with GRU	**Model: "sequential_8"** Layer (type) / Output Shape / Param # conv3d_34 (Conv3D) — (None, 25, 120, 120, 16) — 1,312 batch_normalization_30 (BatchNormalization) — (None, 25, 120, 120, 16) — 64 conv3d_35 (Conv3D) — (None, 25, 120, 120, 16) — 6,928 batch_normalization_31 (BatchNormalization) — (None, 25, 120, 120, 16) — 64 max_pooling3d_20 (MaxPooling3D) — (None, 12, 60, 60, 16) — 0 conv3d_36 (Conv3D) — (None, 12, 60, 60, 32) — 13,856 batch_normalization_32 (BatchNormalization) — (None, 12, 60, 60, 32) — 128 conv3d_37 (Conv3D) — (None, 12, 60, 60, 32) — 27,680 batch_normalization_33 (BatchNormalization) — (None, 12, 60, 60, 32) — 128 max_pooling3d_21 (MaxPooling3D) — (None, 6, 30, 30, 32) — 0 conv3d_38 (Conv3D) — (None, 6, 30, 30, 64) — 55,360 batch_normalization_34 (BatchNormalization) — (None, 6, 30, 30, 64) — 256 conv3d_39 (Conv3D) — (None, 6, 30, 30, 64) — 110,656 batch_normalization_35 (BatchNormalization) — (None, 6, 30, 30, 64) — 256 max_pooling3d_22 (MaxPooling3D) — (None, 6, 15, 15, 64) — 0 conv3d_40 (Conv3D) — (None, 4, 13, 13, 128) — 221,312 conv3d_41 (Conv3D) — (None, 2, 11, 11, 128) — 442,496 max_pooling3d_23 (MaxPooling3D) — (None, 2, 5, 5, 128) — 0 global_average_pooling3d_1 (GlobalAveragePooling3D) — (None, 128) — 0 dense_23 (Dense) — (None, 128) — 16,512 batch_normalization_36 (BatchNormalization) — (None, 128) — 512 dropout_25 (Dropout) — (None, 128) — 0 dense_24 (Dense) — (None, 512) — 66,048 batch_normalization_37 (BatchNormalization) — (None, 512) — 2,048 dense_25 (Dense) — (None, 5) — 2,565 **Total params:** 968,181 (3.69 MB) **Trainable params:** 966,453 (3.69 MB) **Non-trainable params:** 1,728 (6.75 KB)

<table>
<tr>
<td>8</td>
<td>Time Conv2D + GRU</td>
<td>Accuracy: 85.1
Val Accuracy: 79.5</td>
<td>Explanation - Model is learning without overfitting

Decision - Try Conv2D with LSTM</td>
<td>Model: "sequential"
<table>
<tr><th>Layer (type)</th><th>Output Shape</th><th>Param #</th></tr>
<tr><td>time_distributed (TimeDistributed)</td><td>(None, 25, 120, 120, 32)</td><td>896</td></tr>
<tr><td>batch_normalization (BatchNormalization)</td><td>(None, 25, 120, 120, 32)</td><td>128</td></tr>
<tr><td>time_distributed_1 (TimeDistributed)</td><td>(None, 25, 60, 60, 32)</td><td>0</td></tr>
<tr><td>time_distributed_2 (TimeDistributed)</td><td>(None, 25, 60, 60, 64)</td><td>18,496</td></tr>
<tr><td>batch_normalization_1 (BatchNormalization)</td><td>(None, 25, 60, 60, 64)</td><td>256</td></tr>
<tr><td>time_distributed_3 (TimeDistributed)</td><td>(None, 25, 30, 30, 64)</td><td>0</td></tr>
<tr><td>time_distributed_4 (TimeDistributed)</td><td>(None, 25, 30, 30, 128)</td><td>73,856</td></tr>
<tr><td>batch_normalization_2 (BatchNormalization)</td><td>(None, 25, 30, 30, 128)</td><td>512</td></tr>
<tr><td>time_distributed_5 (TimeDistributed)</td><td>(None, 25, 15, 15, 128)</td><td>0</td></tr>
<tr><td>time_distributed_6 (TimeDistributed)</td><td>(None, 25, 128)</td><td>0</td></tr>
<tr><td>time_distributed_7 (TimeDistributed)</td><td>(None, 25, 128)</td><td>16,512</td></tr>
<tr><td>batch_normalization_3 (BatchNormalization)</td><td>(None, 25, 128)</td><td>512</td></tr>
<tr><td>dropout (Dropout)</td><td>(None, 25, 128)</td><td>0</td></tr>
<tr><td>gru (GRU)</td><td>(None, 128)</td><td>99,072</td></tr>
<tr><td>batch_normalization_4 (BatchNormalization)</td><td>(None, 128)</td><td>512</td></tr>
<tr><td>dense_1 (Dense)</td><td>(None, 5)</td><td>645</td></tr>
</table>
Total params: 211,397 (825.77 KB)
Trainable params: 210,437 (822.02 KB)
Non-trainable params: 960 (3.75 KB)</td>
</tr>
<tr>
<td>9</td>
<td>Time Conv2D + ConvLSTM</td>
<td>Accuracy: 61.8
Val Accuracy: 55.4</td>
<td>Explanation - Model is not able to learn enough, but there is no sign of overfitting

Decision - GRU is giving better accuracy and is an equivalently simpler model</td>
<td>Model: "sequential_1"
<table>
<tr><th>Layer (type)</th><th>Output Shape</th><th>Param #</th></tr>
<tr><td>time_distributed_8 (TimeDistributed)</td><td>(None, 25, 120, 120, 32)</td><td>896</td></tr>
<tr><td>batch_normalization_5 (BatchNormalization)</td><td>(None, 25, 120, 120, 32)</td><td>128</td></tr>
<tr><td>time_distributed_9 (TimeDistributed)</td><td>(None, 25, 60, 60, 32)</td><td>0</td></tr>
<tr><td>time_distributed_10 (TimeDistributed)</td><td>(None, 25, 60, 60, 64)</td><td>18,496</td></tr>
<tr><td>batch_normalization_6 (BatchNormalization)</td><td>(None, 25, 60, 60, 64)</td><td>256</td></tr>
<tr><td>time_distributed_11 (TimeDistributed)</td><td>(None, 25, 30, 30, 64)</td><td>0</td></tr>
<tr><td>time_distributed_12 (TimeDistributed)</td><td>(None, 25, 30, 30, 128)</td><td>73,856</td></tr>
<tr><td>batch_normalization_7 (BatchNormalization)</td><td>(None, 25, 30, 30, 128)</td><td>512</td></tr>
<tr><td>time_distributed_13 (TimeDistributed)</td><td>(None, 25, 15, 15, 128)</td><td>0</td></tr>
<tr><td>conv_lstm2d (ConvLSTM2D)</td><td>(None, 13, 13, 16)</td><td>83,008</td></tr>
<tr><td>batch_normalization_8 (BatchNormalization)</td><td>(None, 13, 13, 16)</td><td>64</td></tr>
<tr><td>global_average_pooling2d_1 (GlobalAveragePooling2D)</td><td>(None, 16)</td><td>0</td></tr>
<tr><td>dense_2 (Dense)</td><td>(None, 128)</td><td>2,176</td></tr>
<tr><td>batch_normalization_9 (BatchNormalization)</td><td>(None, 128)</td><td>512</td></tr>
<tr><td>dropout_1 (Dropout)</td><td>(None, 128)</td><td>0</td></tr>
<tr><td>dense_3 (Dense)</td><td>(None, 512)</td><td>66,048</td></tr>
<tr><td>batch_normalization_10 (BatchNormalization)</td><td>(None, 512)</td><td>2,048</td></tr>
<tr><td>dropout_2 (Dropout)</td><td>(None, 512)</td><td>0</td></tr>
<tr><td>dense_4 (Dense)</td><td>(None, 5)</td><td>2,565</td></tr>
</table>
Total params: 250,565 (978.77 KB)
Trainable params: 248,805 (971.89 KB)
Non-trainable params: 1,760 (6.88 KB)</td>
</tr>
</table>

Selected Model

Model Selected - Time Distributed Conv2D + GRU
Accuracy = 80.51
Validation Accuracy = 81.25

Hyperparameters:

- batch_size = 16
- learning_rate = 0.01
- epochs = 50