**GESTURE RECOGNITION**

**3D CNN model**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Experiment Number** | **Model** | **Model Attributes/Parameters** | **Result** | **Total # of Parameters** | **Decision + Explanation** |
| 1 | Model 1 | * No. Convolution Layers - 6 * Convolution Filter Size- (2,2,2) * Pooling filter size - (2,2,2) * Batch Size - 15 * Epochs - 10 * Image size - (120, 120) | * Model is overfitting * Train loss: 0.8216 * Train categorical accuracy: 0.7210 * Val loss: 1.2255 * Val categorical accuracy: 0.6100 | Total params:  3,370,093  Trainable params: 3,369,821  Non-trainable params: 272 | Clearly, model is overfitting. Let’s drop some of the neurons on dense layer in model 2 and increase the epochs to 25 |
| 2 | Model 2 | * No. Convolution Layers - 6 * Convolution Filter Size- (2,2,2) * Pooling filter size - (2,2,2) * Batch Size - 15 * Epochs - 25 * Image size - (120, 120) | * Model is overfitting * Train loss: 0.5430 * Train categorical accuracy: 0.7768 * Val loss: 1.1338 * Val categorical accuracy: 0.6900 | Total params:  3,303,021  Trainable params: 3,302,749  Non-trainable params: 272 | Clearly, model is overfitting. Lets further drop some of the neurons on dense layer in model 3 |
| 3 | Model 3 | * No. Convolution Layers - 6 * Convolution Filter Size- (2,2,2) * Pooling filter size - (2,2,2) * Batch Size - 15 * Epochs - 25 * Image size - (120, 120) | * Train loss: 0.3063 * Train categorical   accuracy: 0.9005   * Val loss: 0.8488 * Val categorical accuracy: 0.8200 | Total params:  1,464,301  Trainable params: 1,463,261  Non-trainable params: 1,040 | This is a stable model and it is not overfitting.  Even accuracy for train and Valid set is good. |

**2D CNN + RNN(GRU/LSTM)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Experiment Number** | **Model** | **Model Parameters** | **Result** | **Total of Parameters** | **Decision + Explanation** |
| 1 | Model 1 | No. Convolution Time Distributed Layers : 1 (64 convolutions)  Convolution Filter Size : (3,3)  - Pooling filter size : (2,2)  GRU : 64  Dense : 100  Batch Size : 50  Epochs : 1  Image size : (70, 70)  Number of images per video : 30 | * Train loss: 1.6495 * Train categorical accuracy: 0.1825 * Val loss: 1.6119 * Val categorical accuracy: 0.2000 | Total params:  14,226,397  Trainable params:  14,226,397  Non-trainable params: 0 | Model has very low accuracy.  We will create new model with additional conv2d time distributed layers, Decreased batch size with increased number of layers. |
| 2 | Model 2 | No. Convolution TimeDistributed Layers : 4 (64-32-64-64 convolutions)  Dropouts – 0.25  Convolution Filter Size : (3,3)  - Pooling filter size : (2,2)  GRU : 64  Dense : 100  Batch Size : 30  Epochs : 10  Image size : (70, 70)  Number of images per video : 30 | * Train loss: 0.2903 * Train categorical accuracy: 0.9020 * Val loss: 0.6828 * Val categorical accuracy: 0.7500 | Total params: 3,046,653  Trainable params:  3,046,205  Non-trainable params:  448 | The model is overfitting.we will increase the conv time distributed layer to 6 for next model. |
| 3 | Model 3 | No. Convolution TimeDistributed Layers : 6 (64-32-64-64-128-128 convolutions)  Dropouts – 0.25  Convolution Filter Size : (3,3)  - Pooling filter size : (2,2)  GRU : 64  Dense : 512  Batch Size : 30  Epochs : 10  Image size : (70, 70)  Number of images per video : 30 | * Train loss: 7.3055 * Train categorical accuracy: 0.9367 * Val loss: 7.7596 * Val categorical accuracy: 0.7600 | Total params: 2,182,565 Trainable params: 2,181,605 Non-trainable params: 960 | This model gave better val\_accuracy but still has overfitting issue. We will build new model by decreasing the learning rate using callbacks(patience =1) |
| 4 | Model 4 | No. Convolution Time Distributed Layers : 6 (64-32-64-64-128-128 convolutions)  Dropouts – 0.25  Convolution Filter Size : (3,3)  - Pooling filter size : (2,2)  GRU : 64  Dense : 512  Batch Size : 20  Epochs : 20  Image size : (70, 70)  Number of images per video : 30 | * Train loss: 6.6168 * Train categorical accuracy: 0.9985 * Val loss: 7.0530 * Val categorical accuracy: 0.8300 | Total params:  2,181,413 | Model is still overfitting. So we will use transfer learning in the next model  Decreased learning rate with patience = 1 and also decreased the batch size to 20 and tried to run for more epochs to 20 to see if model is converging. |
| 5 | Model 5 | Resnet50 with below 150 layers non trainable  GRU : 1000  Time distributed Dense : 100  Dense:64  Batch Size : 20  Epochs : 15  Image size : (197, 197)  Number of images per video : 20 | * Train loss: 0.0312 * Train categorical accuracy: 0.9970 * Val loss: 0.0513 * Val categorical accuracy: 0.9900 | Total params: 32,966,201  Trainable params: 19,368,633  Non-trainable params: 13,597,568 | Resnet50 model has managed the overfitting well. Transfer learning has been successful. Let’s build another model to see if we can further improve the model with 140 layers. |
| 6 | Model 6 | Resnet50 with below 140 layers non trainable  LSTM : 1024  Time distributed Dense : 128  Dense:64  Batch Size : 20  Epochs : 25  Image size : (197, 197)  Number of images per video : 20 | * Train loss: 0.0050 * Train categorical accuracy: 1.0000 * Val loss: 0.0088 * Val categorical accuracy: 1.0000 | Total params: 36,470,149  Trainable params: 27,860,485  Non-trainable params: 8,609,664 | We changed GRU layer to LSTM and fitted a model with perfect accuracy for train and validation dataset in last epoch and even managed to reduce the loss further. |