## **Gesture Recognition Project**

|  |  |  |  |
| --- | --- | --- | --- |
| **Exp No.** | **Model** | **Result** | **Decision + Explanation** |
| 1 | Conv3D(32)  MaxPooling3D  BatchNormalization  Conv3D(64)  MaxPooling3D  BatchNormalization  Conv3D(128)  MaxPooling3D  BatchNormalization  Dropout(0.5)  Adam(LR=0.001)  **No. of frames:** 10  **Image Size:** 120 x 160  **Batch Size:** 16 | categorical\_accuracy: **0.9999**  loss: 0.0199  val\_categorical\_accuracy: **0.93**  val\_loss: 0.1590 | Multiple dips in the accuracy curve  Validation accuracy fluctuates significantly and does not follow a consistent improvement trend.  Validation loss is high in compared to train loss  **For Next Model**: Remove 2 Normalization  Add 2 more dropout layers |
| 2 | Conv3D(32)  MaxPooling3D  Dropout(0.2)  Conv3D(64)  MaxPooling3D  Dropout(0.2)  Conv3D(128)  MaxPooling3D  BatchNormalization  Dropout(0.5)  Adam(LR=0.001)  **No. of frames:** 10  **Image Size:** 120 x 160  **Batch Size:** 16 | categorical\_accuracy: **0.9516**  loss: 0.2946  val\_categorical\_accuracy: **0.6700**  val\_loss: 2.0743 | This model seems to be overfitting as the accuracies are wide apart.  **For Next Model:**  Reduce trainable paras by reducing kernels and LR |
| 3 | Conv3D(16)  MaxPooling3D  Dropout(0.2)  Conv3D(32)  MaxPooling3D  Dropout(0.2)  Conv3D(64)  MaxPooling3D  BatchNormalization  Dropout(0.5)  Dense(128)  Adam(LR=0.0005)  **No. of frames:** 10  **Image Size:** 120 x 160  **Batch Size:** 16 | categorical\_accuracy: **0.9397**  loss: 0.2972  val\_categorical\_accuracy: **0.760**  val\_loss: 0.989 | Validation Loss increased from earlier model.  Reduced parameters did not help as much.  **For Next Model:**  Go back to 1st architecture with reduced LR. |
| 4 | Conv3D(32)  MaxPooling3D  Conv3D(64)  MaxPooling3D  Conv3D(128)  MaxPooling3D  BatchNormalization  Dropout(0.5)  Dense(128)  Dropout(0.25)  Adam(LR=0.0005)  **No. of frames:** 10  **Image Size:** 120 x 160  **Batch Size:** 16 | categorical\_accuracy: **0.9869**  loss: 0.0401  val\_categorical\_accuracy: **0.720**  val\_loss: 3.1581 | The validation loss has increased and also we can observe that the training accuracy was retained as before, but the validation accuracy has dropped.  **For Next Model:**  Add Normalization and dropout layers and increase model complexity  Let’s try LSTM architectures. |
| 5 | Conv3D(32)  MaxPooling3D  BatchNormalization  Dropout(0.25)  Conv3D(64)  MaxPooling3D  BatchNormalization  Dropout(0.25)  Dense(64)  Dropout(0.5)  Adam(LR=0.0002)  **No. of frames:** 10  **Image Size:** 120 x 160  **Batch Size:** 16 | categorical\_accuracy: **0.9371**  loss: 0.1638  val\_categorical\_accuracy: **0.7500**  val\_loss: 1.7131 | Although, Validation loss has dropped, so do the accuracies. Still overfitting.  **For Next Model:**  Let’s try LSTM architectures. |
| 6 | TD Conv2D(16)  TD MaxPooling2D  TD Conv2D(32)  TD MaxPooling2D  TD Flatten  **LSTM**(64)  Adam(LR=0.0002)  **No. of frames:** 10  **Image Size:** 120 x 160  **Batch Size:** 16 | categorical\_accuracy: **1.0000**  loss: 0.0032  val\_categorical\_accuracy: **0.8300**  val\_loss: 0.8103 | Although, Validation loss improved, model is overfitting  Val Accuracy increased in epoch 20 but Train accuracy reach 1  **For Next Model:**  Add Normalization and dropout layers and increase model complexity |
| 7 | TD Conv2D(32)  TD MaxPooling2D  TD BatchNormalization  TD Conv2D(64)  TD MaxPooling2D  TD BatchNormalization  TD Conv2D(128)  TD MaxPooling2D  TD BatchNormalization  TD Flatten  **LSTM**(128)  Adam(LR=0.0005)  **No. of frames:** 10  **Image Size:** 120 x 160  **Batch Size:** 16 | categorical\_accuracy: **0.6823**  loss: 1.2175  val\_categorical\_accuracy: **0.4300**  val\_loss: 1.3868 | Accuracies are low. Model took 3+ hours to run.  Let’s try **GRU** architectures  **For Next Model:**  Lets reduce the kernels, add dropouts and increase LR. |
| 8 | TD Conv2D(16)  TD MaxPooling2D  TD BatchNormalization  TD Dropout(0.2)  TD Conv2D(32)  TD MaxPooling2D  TD BatchNormalization  TD Dropout(0.2)  TD Conv2D(64)  TD MaxPooling2D  TD BatchNormalization  TD Flatten  **GRU**(64, dropout=0.25)  Adam(LR=0.001)  **No. of frames:** 10  **Image Size:** 120 x 160  **Batch Size:** 16 | categorical\_accuracy: **1.0000**  loss: 0.0052  val\_categorical\_accuracy: **0.7100**  val\_loss: 0.8661 | Model is overfitting.  Too many parameters.  **For Next Model:**  1. Increasing dropout  2. Adding Global Pooling Instead of Flatten  3. learning\_rate 0.0001. |
| 9 | TD Conv2D(16)  TD MaxPooling2D  TD BatchNormalization  TD Dropout(0.3)  TD Conv2D(32)  TD MaxPooling2D  TD BatchNormalization  TD Dropout(0.2)  TD Conv2D(64)  TD MaxPooling2D  TD BatchNormalization  GlobalAveragePooling2D  **GRU**(32, dropout=0.3)  Adam(LR=0.0001)  **No. of frames:** 10  **Image Size:** 120 x 160  **Batch Size:** 16 | categorical\_accuracy: **0.7491**  loss: 0.7260  val\_categorical\_accuracy: **0.510**  val\_loss: 1.0346 | Low accuracies with overfitting.  **For Next Model:**  Resize the image to square matrix 120X120. Increase no. of frames and decreased batch size. |
| 10 | Conv3D(32)  BatchNormalization  MaxPooling3D  Conv3D(64)  BatchNormalization  MaxPooling3D  Conv3D(128)  BatchNormalization  MaxPooling3D  Conv3D(256)  BatchNormalization  MaxPooling3D  Dense(128)  BatchNormalization  Dropout(0.25)  Dense(128)  BatchNormalization  Dropout(0.25)  **No. of frames:** 20  **Image Size:** 120 x 120  **Batch Size:** 10 | categorical\_accuracy: **0.9744**  loss: 0.0915  val\_categorical\_accuracy: **0.9600**  val\_loss: 0.1851 | There are dips in validation accuracy and spikes in validation loss. Towards the last epochs, metrices seem good in this model.  **For Next Model:**  Let’s try transfer learning. |
| 11 | TD(MobileNet)  BatchNormalization  MaxPooling2D  LSTM(128)  Dropout(0.25)  Dense(128)  Dropout(0.25)  **No. of frames:** 20  **Image Size:** 120 x 120  **Batch Size:** 10 | categorical\_accuracy: **1.0000**  loss: 3.3689e-04  val\_categorical\_accuracy: **0.9380**  val\_loss: 0.3951 | Accuracies are good but the validation metrices are unstable with lot of dips and spikes.  **For Next Model:**  Let’s try GRU with reduced frames. |
| 12 | TD(MobileNet)  BatchNormalization  MaxPooling2D  GRU(128)  Dropout(0.25)  Dense(128)  Dropout(0.25)  **No. of frames:** 15  **Image Size:** 120 x 120  **Batch Size:** 10 | categorical\_accuracy: **0.9774**  loss: 0.0868 val\_categorical\_accuracy: **0.9600**  val\_loss: 0.1258 | Seems to be best model by far.  **For Next Model:**  Let’s try ResNet with LSTM |
| 13 | TD(ResNet)  BatchNormalization  MaxPooling2D  LSTM(128)  Dropout(0.25)  Dense(128)  Dropout(0.25)  **No. of frames:** 20  **Image Size:** 120 x 120  **Batch Size:** 10 | categorical\_accuracy: **0.8100**  loss: 0.5075 val\_categorical\_accuracy: **0.5500**  val\_loss: 1.2701 | Model overfits.  **For Next Model:**  Let’s try ResNet with GRU |
| 14 | TD(ResNet)  BatchNormalization  MaxPooling2D  GRU(128)  Dropout(0.25)  Dense(128)  Dropout(0.25)  **No. of frames:** 20  **Image Size:** 120 x 120  **Batch Size:** 10 | categorical\_accuracy: **0.8175**  loss: 0.5145 val\_categorical\_accuracy: **0.5700**  val\_loss: 1.4842 | Model still overfits.  **For Next Model:**  Let’s increase the number of frames in model 12 |
| 15 | TD(MobileNet)  BatchNormalization  MaxPooling2D  GRU(128)  Dropout(0.25)  Dense(128)  Dropout(0.25)  **No. of frames:** 20  **Image Size:** 120 x 120  **Batch Size:** 10 | categorical\_accuracy: **0.9864**  loss: 0.0373  val\_categorical\_accuracy: **0.980**  val\_loss: 0.0898 | Model accuracy has improved. |
| **Final Model** | TD(MobileNet)  BatchNormalization  MaxPooling2D  GRU(128)  Dropout(0.25)  Dense(128)  Dropout(0.25)  **No. of frames:** 20  **Image Size:** 120 x 120  **Batch Size:** 10 | categorical\_accuracy: **0.9864**  loss: 0.0373  val\_categorical\_accuracy: **0.980**  val\_loss: 0.0898 | This model is selected as the final model, as we have got train accuracy 98.6% and 98% validation accuracy. |

**Table 1: Observations and Results**