**GESTURE RECOGNITION WRITE-UP**

**Giridhar Challa**

**Keerthana S**

**Nirakar Nayak**

**Batch size selection:** Starting with batch size of 128 caused the OOM error. Experimented with multiple batch sizes and got better results with batch size 10. As the batch size is 10, some amount of extra time is consumed .

**Image resolution selection:** The image resolution less than 80x80 is leading to undetermined results. After multiple experiments the image resolution that are selected for modelling are 140x140 and 160x160 which are working good with the models. Experimented with both resolutions. When using LSTM models, 140x140 resolution produced better results.

The first model is to check the optimal resource utilization with higher batch size-128 and frames to process as 30 lead to “**Original error: RESOURCE\_EXHAUSTED: Out of memory while trying to allocate 6308233216 bytes.”**

So the rest of the models are chosen wisely based on multiple parameters like batch size, image resolution, how many images to be selected from each folder, epochs, filters while using Conv3D models, LSTM and Transfer learning models.

We are using Adam optimizer as the model adapts learning rate and faster convergence. We expect it to perform well with this conv3D model as well.

**NOTE::** The notebook is developed in Kaggle where the model files are stores as **.keras** instead of **.h5** which is deprecated in latest keras and tf versions. In case to generate model files with .h5 extension we need to execute in Jupyter notebook or colab.

We need to change the filepath in the script with keras when used in Kaggle and h5 when used in jupyter notebook.

“””filepath = model\_name + 'model-{epoch:05d}-{loss:.5f}-{categorical\_accuracy:.5f}-{val\_loss:.5f}-{val\_categorical\_accuracy:.5f}.keras'”””

“””filepath = model\_name + 'model-{epoch:05d}-{loss:.5f}-{categorical\_accuracy:.5f}-{val\_loss:.5f}-{val\_categorical\_accuracy:.5f}.h5”””

**Idea behind selecting the Final model layers and Parameters:**

* Better results are produced while using higher image resolution in the models. Image resolution used in the final model is 140x140. Combining dropouts with batch normalization mitigated the overfitting issues and enhanced the generalization of the models.
* Using Adam optimizer the learning rates are more adaptive when the loss is not decreasing and thus slowing rate of learning helped in improving the accuracies.
* Final model trainable parameters:
  + **Total params: 5,413,317**
  + **Trainable params: 2,182,405**
  + **Non-trainable params: 3,230,912**
* Transfer learning has significantly increased the accuracy of the model from overfitting to the best model. There is a huge shift from lower accuracy to more than 90% accuracy in both training and validation data.
* Used image resolution with 140x140 and batch size=10 to learn the patterns in the data effectively.
* With lesser epochs itself the best results are obtained.

**In the final code file we commented all the codes that experimented except final codes so that the final code works good. The outputs of each cell is saved.**

**Different Models Tested:**

|  |  |  |  |
| --- | --- | --- | --- |
| **EXPERIMENT** | **MODEL** | **RESULT** | **DECISION+EXPLANATION** |
| **1** | **Conv3D** | **Error** | **With the batch size 128, the resources are exhausted. So planned to reduce the batch size and adjust the other parameters to train the models**  **Original error: RESOURCE\_EXHAUSTED: Out of memory while trying to allocate 6308233216 bytes.** |
| **2** | **Conv3D** | **Training Accuracy=<0.6**  **Val Accuracy<0.60** | **Method: Starting with 3 conv layers and FC layer with dropout and BN at each layer**  **The model is not performing well for 120x120 and 140x140 with batch\_size=30 frames\_sample=20 or 25 and there is lot differences between training and validation accuracies.**  **Needs to try with less frames\_to\_sample and more image resolution**  **The model didn’t perform well so we need to increase the layers**  **Needs to add more conv layers** |
| **3-1**  **3-2** | **Conv3D** | **For 140x140 resolution:**  **Training Accuracy=0.7188**  **Val Accuracy=0.82**  **For 160x160 resolution:**  **Training Accuracy=0.8322**  **Val Accuracy=0.83** | **Method: With no dropouts in 4 conv layers and dropouts in FC layers. Used BN in all layers with *Maxpooling***  **Tried to build 2 models with resolution 140x140 and 160x160**  **BS=10, Frames to Sample 20 and epochs 20**  **The model performed well compared to previous.**  **Needs to try with same model with reduced batch size and increased epochs**  **Planned to try with both resolutions by changing the Pooling layers.** |
| **4-1**  **4-2** | **Conv3D** | **For 140x140 resolution:**  **Training Accuracy=0.7966**  **Val Accuracy=0.77**  **(At Epoch 19/20)**  **For 160x160 resolution:**  **Training Accuracy=0.8359**  **Val Accuracy=0.72** | **Method: With no dropouts in 4 conv layers and dropouts in FC layers. Used BN in all layers with *AveragePooling***  **Tried with resolution 140x140 BS=10, Frames to Sample 10 and epochs 20**  **The model performed well compared to previous. All the inner lying features are correctly captured**  **Needs to try with same model with reduced batch size and increased epochs. There is a steady improvement. Planned to check with reduced batch size and check the performance.**  **Tried with resolution 160x160 BS=10, Frames to Sample 10 and epochs 20**  **By changing the pooling method the model didn’t performed well compared to previous.** |
| **5** | **Conv3D** | **Training Accuracy=0.4565**  **Val Accuracy=0.63** | **Method: Same as above method with reduced batch size as 5**  **With resolution 140x140 BS=5, the model didn’t performed well.**  ***So planned to go with batch size as 10.***  **Also planned to add dropouts on conv layers** |
| **6** | **Conv3D** | **Training Accuracy=0.7076**  **Val Accuracy=0.78** | **Method: Adding dropouts=0.25 after conv layers and 0.5 after FC layers for the above model. Resolution 140x140**  **The model is not performing well on training data.**  **Planned to decrease the dropout.** |
| **7** | **Conv3D** | **Training Accuracy=0.90**  **Val Accuracy=0.74** | **Method: Changing to 0.25 after FC layers for the above model. Resolution 140x140**  **The model is performing well on training data. But the model is overfitting**  **Planned to increase the dropout.** |
| **8** | **Conv3D** | **Training Accuracy= 0.7450 Val Accuracy=0.76** | **Method: Changing to 0.4 after FC layers for the above model. Resolution 140x140**  **The training accuracy is reduced. The accuracies with dropout=0.4 value is not good.**  ***Planned to use dropout as 0.25*** |
| **9** | **CNN+LSTM** | **Training\_Accuracy=**  **0.9384 Val\_Accuracy=0.85**  **(At Epoch 22/25)**  **Training\_Accuracy=**  **0.96 Val\_Accuracy=0.85**  **(At Epoch 22/25)** | **With 140x140 resolution, the model is overfitting But got the best accuracies at Epoch 22/25.**  **After 25 Epochs: Training Accuracy=0.9897 Val Accuracy=0.80**  **With 160x160 resolution, the model is overfitting but got the best accuracies at Epoch 22/25.**  **After 25 Epochs: Training Accuracy=0.9779 Val Accuracy=0.84**  **Overfiting is less in case of 140x140 resolution.** |
| **10** | **CNN+LSTM with GRU** | **Training\_Accuracy=**  **0.8861 Val\_Accuracy=0.81**  **(At Epoch 22/25)** | **With 140x140 resolution, the model is overfitting But got the best accuracies at Epoch 22/25.**  **After 25 Epochs: Training Accuracy=0.9019 Val Accuracy=0.80**  **We see the better modelling at Epoch 22.** |
| **11** | **ConvD with Data Augmentation on Model3** | **Training Accuracy=0.8865 Val Accuracy=0.65** | **With 140x140 resolution, the model is overfitting with data augmentation** |
| **12** | **ConvD with Data Augmentation on Model4** | **Training Accuracy=0.8592 Val Accuracy=0.74** | **With 140x140 resolution, the model is overfitting with data augmentation**  **Overfitting is reduced.** |
| **13** | **Transfer Learning with LSTM with Augmentation** | **Dropouts:0.5 with Augmentation:**  **Training Accuracy=0.9653**  **Val Accuracy=0.90** | **Used MobileNet architecture with LSTM.**  **Epochs:10 Resolution 140x140 LSTMcells:128**  **Dropouts:0.5 with Augmentation** |
| **14** | **Transfer Learning with LSTM without Augmentation**  **[[FINAL MODEL]]** | **Dropouts:0.4 without Augmentation:**  **Training Accuracy=0.9367**  **Val Accuracy=0.90**  **Epoch5/10** | **Used MobileNet architecture with LSTM.**  **Epochs:10 Resolution 140x140 LSTMcells:128**  **Dropouts:0.4 without Augmentation**  **The best output is obtained with Epoch5/10 which has the best model.**  **The validation loss is reduced significantly in the model**  **We are considering this as the best model with Epoch5 results.** |

**The weights of the best model is “model\_init\_2025-03-1717\_07\_07.960764\model-00005-0.22208-0.93665-0.26060-0.90000.h5”**