First a conv3D model is bult and conducted experiments. Final model is run and saved based on better accuracy.

* Fifteen frames from a video are chosen in an alternate fashion. Taking half of the images with even index numbers will also give a sequence that progresses with gestures
* Trying 2 resolutions 100 x 100 and 120 x 120
* As it is a multi-classification problem, categorical accuracy is the metric used.
* Best model is saved based on val\_categorical\_accuracy(maximise)
* Categorical accuracy is used as metric as we have multi classification problem.
* Learning rate decay done using val loss

Then built a conv2D-GRU model.

* We will use TimeDistributed layer to pass each frame of a video to Conv2D in a sequential manner.
* The output of Conv2D layer is going to be inputted to GRU. Here we are using GRU because it is lighter, and we think GRU will be sufficient for this hand gesture recognition task.
* We will also make use of Transfer learning (VGG16) rather than a custom ConvNet for better result. Last 3 layers of VGG16 is made as trainable to make our network specific to our problem.
* The output layer will have a softmax function to multi-class classification
* Experiments are mainly conducted to get a stable model

|  |  |  |  |
| --- | --- | --- | --- |
| Experiment Number | Model | Result | Decision + Explanation |
| 1 | Conv3D | Unable to create model | Pooling filter changed to (1,2,2) from (2,2,2) in last layers as the output dimension went negative in case of dimension representing frames |
| 2 | Conv3D | Space error | Cleared some space also changed the code to save only best model |
| 3 | Conv3D | Checked whether model is working | Model working |
| 4 | Conv3D | Model is overfitting | Model overfitted for 20 epochs. Model is learning. Here best model is saved based on the validation loss. But better result is given based on val accuracy. Even if val loss is decreasing, sometimes vali accuracy won’t increase. Choosing val\_categorical\_accuracy for saving best model will make sense here. |
| 5 | Conv3D | Batch size 512 threw oom error | Experimenting with batch size 128 |
| 6 | Conv3D | Batch size 128 threw oom error | Experimenting with batch size 64 |
| 7 | Conv3D | Batch size 64 works for our set of data and GPU. | Batch size fixed as 64 as batch running fine with 64. |
| 8 | Conv3D | Image size of 120x120 and 100x100 ran for 5 epochs | There is no difference in the runtime and accuracy is better for 120x120 image size. So choosing to go with 120x120 |
| 9 | Conv3D | Accuracy: 0.65  (model name- model) | Try ConvLSTM to see if it can give better categorical accuracy |
| 10 | ConvLSTM | Loss stagnates | Increased GRU units from 64 to 100 and reduced decision layers to decrease number of trainable parameters. |
| 11 | ConvLSTM | Loss is unstable | Increased units to 120. Will try the ablation run with more train data, batch size of 64 for 10 epochs. This may stabilize the model. |
| 12 | ConvLSTM | Accuracy=0.24  Loss reducing overall | Loss is reducing overall but still model seems to be unstable. One more GRU layer was added to see if our accuracy is improving. Also training with more samples and larger epoch may create a stable model. |
| 13 | ConvLSTM | Accuracy=0.24 | Couldn’t stabilise model |