# Writeup on the Gesture Recognition model building activity.

## Building multiple models and arriving at the right model and architecture

In this project we have fully focused on the most popular architecture the convolution 3D architecture We have given a try on the CNN/RNN Arch and captured all the results

Among the Models and its variations and with the experimentation with various Hyper parameters and we have arrived at Model 8 3D Conv 3D as the most suitable (**3DConv_Model_8_Neural_Networks_Gesture_Recognition_FINAL_SELECT.ipynb**) based on Conv3D Architecture as our final model.

Key parameters of the working model (Conv 3d – Model No 8 below) that was experimented and arrived at is captured here.

In addition, All the other Conv3D Models and variations too were experimented with different values of batch size and epoch values to arrive at a good accuracy. Below the summary write-up

| ModelNo | Model Arch | Result | Decision + Explanation |
|---|---|---|---|
| | | | |
| 1 | Conv3D | Huge trainable parameters. Takes a lot of time in training and not a great accuracy. Accuracy in the range .40 to .60 | -Only very minimal and simple processing of Resizing and normalizing is done is done in this model<br><br>Action:<br>-Perform comprehensive preprocessing using standard libraries to make the data proper. |
| | | | |
| 2 | Conv3D | Still there is a huge trainable parameter Takes a lot of time in training and not a great accuracy. Accuracy in the range of .40 to .60 | Performed transformations, Resize and cropping and normalizing done using the OpenCV library., changed the model<br><br>Action:<br>-Further advanced preprocessing of data, optimizer can be used. New model can be created |
| | | | |
| 3 | Conv3D | Huge trainable parameters Takes a lot of time in training and Low accuracy in the range of 0.1 to 0.3 | -Performed data augmentation, changed model, used optimizers (adam), cropped asymmetric frames, Performed horizontal flip.<br><br>Action:<br>-Create new model/modifications to the model |
| | | | |

| | | | |
|---|---|---|---|
| **4** | **Conv3D** | **Huge trainable parameter<br>Takes a lot of time in training and Low accuracy in the range of 0.1 to 0.3** | **-Changed to a new model and tried it out<br><br>Action:<br>-Try with another model/model variation** |
| | | | |
| **5** | **Conv3D** | **Drastic reduction in trainable parameters and a good level of accuracy.<br>This is also a good model. However, it may be a mildly overfitting model as it often yeilds a has very high accuracy in the range of 0.90 and above** | **-Changed to new model with the following parameters<br>4 Convolutional layer.<br>filter size - (8, 16, 32, 64)<br>kernel size = (3, 3, 3)<br>2 Max Pooling layers<br>Multi-Layer Perceptron<br>Batch normalization on convolution architecture<br>Dense layers and dropout to avoid overfitting<br><br>Action:<br>-Experiment further with hyper parameters and layers to increase accuracy. Try experimenting with batch normalization effect, Dropout effect to reduce overfitting** |
| | | | |
| **6** | **Conv3D** | **Trainable parameters under control.<br>Accuracy is very low in the range of 0.10 to 0.30** | **-Removed batch normalization has a big effect on accuracy and accuracy went down drastically.<br><br>Action:<br>-Add back batch normalization and check if the accuracy is restored** |
| | | | |
| **7** | **Conv3D** | **Accuracy hovers around 0.50 to 0.60 and some occasions in the 0.70** | **Added batch normalization back.<br><br>Action:<br>Accuracy is not great. May be underfitting<br>Try to fine tune other parameters for the increase in accuracy.<br>Too many dropouts are present in model. Try Remove few dropouts to strike a balance between underfitting and good generalization and overfitting** |
| | | | |
| **8** | **Conv3D** | **Accuracy: ranges between 0.80 to 0.90 and consistent on runs with** | **-Removed few dropouts in few layers. This model gives consistent results and high accuracy across all the batch sizes and epochs** |

| | | | |
|---|---|---|---|
| | | **various batch size-epoch combinations** | **-The validation loss consistently reduces fat each epoch most run of the times (for various batch size and epoch combinations)**<br>**-No further improvements may be required.**<br>**This model could be freezed as the FINAL Model** |
| | | | |
| **9** | **CNN-LSTM** | **Low Accuracy** | **-Low Accuracy**<br><br>**Action:**<br>**-We will also try with other models such as GRU and Conv3d**<br>**(Conv3d arch – Model 8 has provided an efficient model as shown above)** |
| | | | |
| **10** | **CNN-GRU** | **Low Accuracy** | **-Low accuracy**<br><br>**Action:**<br>**-We will Try with Conv3d**<br>**(Conv3d arch – Model 8 has provided an efficient model as shown above)** |
| | | | |
| **Final Model** | **Model 8 based on Conv3d** | **Consistent accuracy of more than 80%.** | **Freeze this model as the final model which is based on 3DConvolution Architecture**<br><br>**Model File**<br>**File Name : model-00014-0.31685-0.87908-0.36788-0.88000.h5**<br><br>**Batch Size:20 and No of Epochs 20**<br><br>Epoch 14/2034/34<br>[==============================] - 13s 389ms/step - loss: 0.3169 - categorical_accuracy: 0.8791 - val_loss: 0.3679 - val_categorical_accuracy: 0.8800Epoch 00014: saving model to model_init_2020-03-3011_11_51.849206/model-00014-0.31685-0.87908-0.36788-0.88000.h5 |