**Gesture Recognition Case Study – Krunal Tanna & Shridhar Kawalkar**

Our Problem statement is to recognize Hand Gestures for Smart Television. We have training data for Training purpose which is used to train different We are training different CNN architecture models like Conv3D and ConvLSTM model and we also have validation data on which we have validated the trained models.

Below are the results of the experiment which we performed with its Result and Explanation:

|  |  |  |  |
| --- | --- | --- | --- |
| **Experiment Number** | **Model** | **Parameters** | **Result** |
| 1 | Model - 1| Conv3D | 4 Layers, Batch size = 10,  Epochs = 50 | Categorical\_accuracy = 0.79  Val\_accuracy = 0.77 |
| 2 | Model -2 | Conv3D | 4 Layers, Batch size = 30,  Epochs = 50 | Categorical\_accuracy = 0.87  Val\_accuracy = 0.83 |
| 3 | Model -3 | Conv3D | 4 Layers, Batch size = 50,  Epochs = 15 | Categorical\_accuracy = 0.99  Val\_accuracy = 0.81 |
| 4 | Model -4 | Conv3D | 4 Layers, Batch size = 30,  Epochs=50 | Categorical\_accuracy = 0.97  Val\_accuracy=0.83 |
| 5 | Model – 5 |ConvLSTM | 4 Layers, Batch size = 30,  Epochs = 15 | Categorical\_accuracy = 0.29  Val\_accuracy=0.21 |
| **Final Model** | **Model -2 | Conv3D** | **4 Layers, Batch size = 30,**  **Epochs = 50** | **Categorical\_accuracy = 0.87**  **Val\_accuracy = 0.83** |

**Explanation:**

**For Model-1,** we have used: 1 input and output layer, 4 convolution and max pooling layers to obtain the most important and informatic features, 2 dense layers followed by a Dropout. We have used a kernel size of 3x3 and to standardize the input image size, we have changed it to 120x120.

Here we got max **categorical\_accuracy = 0.79 and val\_accuracy of 0.77**.

Since batch size if small having 10 batches, we tried to increase the batch size to 30, keeping the number of epoch constant for next model.

**For Model-2,** we have used: 1 input and output layer, 4 convolution and max pooling layers to obtain the most important and informatic features, 2 dense layers followed by a Dropout. We have used a kernel size of 3x3 and to standardize the input image size, we have changed it to 120x120.

Here we got max **categorical\_accuracy = 0.87 and val\_accuracy of 0.83**.

Increasing the batch size from 10 to 30, we observed that the accuracy scores are very high, and we found this model very effective. Also, as the difference between the categorical\_accuracy and val\_accuracy is very low, this model is not overfitting and can be proved as a good fit for our usecase.

**For Model-3,** we have used: 1 input and output layer, 4 convolution and max pooling layers to obtain the most important and informatic features, 2 dense layers followed by a Dropout. We have used a kernel size of 3x3 and to standardize the input image size, we have changed it to 120x120.

Here we have tuned the parameters by increasing the batch size to 50 and reducing the number of epochs to 15.

By performing this hyperparameter tuning, we found out the following results:

**categorical\_accuracy = 0.99 and val\_accuracy of 0.81**.

Since the categorical accuracy is very high with almost 100% accuracy and difference between categorical and validation accuracy is high, we find out that this model is overfitting the data and so we decided to discard this model even after giving very high accuracy.

**For Model-4,** we have used: 1 input and output layer, 4 convolution and max pooling layers to obtain the most important and informatic features, 2 dense layers followed by a Dropout. We have used a kernel size of 3x3 and to standardize the input image size, we have changed it to 120x120.

Since for model 3, we were getting overfitted model so we tried to tune the hyper parameters by changing the batch size to 30 and number of iterations to 50. We found the following results:

**categorical\_accuracy = 0.97, val\_accuracy=0.83**

Despite of having a very good accuracy scores, this model also shows a tendency of overfitting, so we discard this model too.

**For Model-5,** we have used: 1 input and output layer, 5 convolution 2D and max pooling 2D layers to obtain the most important and informatic features, 1 LSTM layer and 2 dense layers followed by Dropout. We have used a kernel size of 3x3 and to standardize the input image size, we have changed it to 120x120.

We have used a batch size of 30 and number of iterations = 15. We got the following results:

**categorical\_accuracy = 0.29, val\_accuracy=0.21**

The accuracy scores are very bad, it maybe because of low epochs. So, we ignore this model.