**Hand Gesture Recognition feature development for a smart TV**

**By: Ashutosh Sharma (C36 Batch)**

**SUMMARY:**

In this project following experiments are performed in a sequence to reach the best model:

* Image Data Set Analysis: finding out the image size categories
* Generator code creation
* Model creation: train model code
* Deciding upon initial sample model
  + Following parameters are considered based on some experimentation
    - **Number of frames=30**(have considered all the frames to test) as we will be deliberately considering less number of epochs to train owing to computation time limitations
    - Batch size=32(experimented with batch size 50 but that was not giving better results thus taken this value)
    - No Augmentation
* Testing the sample model to decide upon which image size to consider(here I have experimented with 100x100x3, 120x120x3 and 160x160x3 images respectively)

Based on the result opted **120x120x3** as the image size

* Deciding on conv3D filter(experimented on 2x2x2 and 3x3x3 filters)

Inference: Based on the analysis **3x3x3** filters are considered

* Experimental model with dropouts introduced at each layer

Inference: Model overfitted and hence dropouts were not considered

* Experimental model by introducing different number of Dense layer neurons(here I have experimented with 128 and 64 neurons respectively)

Inference: Not much difference was seen in the performance hence 64 neurons were considered)

* Experimental model with different optimizers : Adam and SGD

Inference: Adam optimizer appeared to be better

* Experimental model with and without augmentation

Inference: Model with Augmentation fared better

* Experimental CNN LSTM model

Inference: The model totally overfitted

* Experimental CNN GRU model

Inference: Model performed better than LSTM but validation stats appear very spiky(probably this can be overcomed by tuning the model parameters better, due to time constraints and computation limitations unfortunately this could not be achieved)

* Experimental RESNET50 LSTM model

Inference: The model appears to perform better than CNN GRU model

* Experimental VGG16 transfer learning model

Inference: The model appears to be the best performing model along with Conv3D model with Augmentation

Best Conv3D model: Model number 11: Model with Augmentation

Best Performing Model: Model Number:15: VGG16 transfer learning model

|  |  |  |  |
| --- | --- | --- | --- |
| **Experiment Number** | **Model** | **Result** | **Decision + Explanation** |
| 1. **Sample Model Creation** | **Conv3D** | **Initially tested with 2 Epochs and batch size 50**  **Later minimized the batch size further to 32 and increased the number of epochs to 5** | **Batch size of 32 and increasing the number of epochs gave better results** |
| 1. **Choosing the image size** | **Conv3D** | **Experimented on 100x100,120x120 and 160x160 image sizes**  **Model numbers:**  **1 to 3** | **Model performance with 120x120 image size performed the best hence 120x120 image size is considered in the subsequent models.Choosing the appropriate filter size.**  Last epoch result when the model is ran for 120x120 image:  Epoch 5: saving model to /content/gdrive/MyDrive/Project\_data/Conv3D\_Test\_2022-08-1916\_51 to 6\_22.069427/model-00005-0.65769-0.76018-1.20458-0.69000.h5  21/21 [==============================] - 85s 4s/step - loss: 0.6577 - categorical\_accuracy: 0.7602 - val\_loss: 1.2046 - val\_categorical\_accuracy: 0.6900 - lr: 0.0010  <keras.callbacks.History at 0x7faab61aae50> |
| 1. **Choosing the appropriate filter size** | **Conv3D** | **Experimented on 3x3x3 and 2x2x2 filters**  **Model Numbers:4 and 5** | **Performance with 3x3x3 filter was better hence this filter will be considered for the subsequent models**  **Accuracy and Loss comparisons:**  **3x3x3 filter:**    **2x2x2 filter:** |
| 1. **Introducing dropouts at each convolution layer** | **Conv3D** | **Introduced dropouts at each convolution layer**  **Model Number: 6** | **Introducing dropouts at each convolution layer did not perform well, hence this implementation is not adopted in the subsequent models** |
| 1. **Model building with different dense neurons** | **Conv3D** | **Introduced dense layer with 128 and 64 neurons respectively**  **Model Numbers:7,8** | **64 Dense neurons model faired comparatively little well hence adopted the configuration for subsequent models**  **128 Dense neuron results:**    **64 dense neurons results:** |
| 1. **Model building with SGD optimizer** | **Conv3D** | **Across all the models till now we had used Adam optimizer**  **Model Number: 9** | **There is not much difference seen in the performance with SGD optimizer hence decided to continue with Adam optimizer in the subsequent models.**  **Adam Optimizer result:**    **SGD Optimizer results:** |
| 1. **Model building with and without augmentation** | **Conv3D** | **Model Numbers: 10,11** | **With Augmentation model appears to be performing well**  **Adopted augmentation to be used in the subsequent models**  **Without Augmentation Results:**    **With Augmentation Results:** |
| 1. **CNN LSTM Model** | **CNN LSTM** | **Model Number:12** | **The model is totally over fitting**  **Results:** |
| 1. **CNN GRU Model** | **CNN GRU** | **Model Number:13** | **The model is somewhat better than LSTM but we can see spikes in the validation loss and accuracies, probably this model can be further tuned with optimized parameters to achieve better results.**  **Results:** |
| 1. **RESNET50 LSTM Model** | **RESNET50 LSTM** | **Model Number:14** | **The model appears to be better than the GRU Model**  **Results:** |
| 1. **VGG16 LSTM** | **VGG16 LSTM** | **Model Number:15** | **The model appears little overfit**  **Results:** |
| **Final Model** | **Conv3D** | **Model with Augmentation**  **Model Number: 11** |  |
| **Final Model** | **Performance Wise** | **VGG16 LSTM**  **Moodel Number: 15** |  |

**H5 files for the two successful models are located at:**

[**https://drive.google.com/drive/folders/1cAkSi2j\_KyZpQWat322GVqyoYz7PJzCU?usp=sharing**](https://drive.google.com/drive/folders/1cAkSi2j_KyZpQWat322GVqyoYz7PJzCU?usp=sharing)