## **Gesture Recognition Assignment**

## Submitted by –

## Manish Kaushal & Himanshu Pal

**Objective**

In this assignment, we have to build a model to recognize 5 hand gestures in a smart television.  We have been given the training data comprising of few hundred videos divided into a sequence of 30 frames. Our task is to train different models on the 'train' folder provided in the dataset to predict the action performed in each sequence / video and which performs well on the 'val' folder as well. Also, the test folder has been withheld for evaluation purposes - your final model's performance will be tested on the 'test' set.

**Architecture Used**

The model has been built based on the 3D Convolution Neural Network architecture. Just like in 2D conv, we move the filter in two directions (x and y), in 3D conv, we move the filter in three directions (x, y and z). In this case, the input to a 3D conv is a video (which is a sequence of 30 RGB images). If we assume that the shape of each image is 120 x 120 x 3, for example, the video becomes a 4D tensor of shape 120 x 120 x 3 x 30 which can be written as (120 x 120 x 30) x 3 where 3 is the number of channels. Hence, deriving the analogy from 2D convolutions where a 2D kernel/filter (a square filter) is represented as (f x f) x c where f is filter size and c is the number of channels, a 3D kernel/filter (a 'cubic' filter) is represented as (f x f x f) x c (here c = 3 since the input images have three channels). This cubic filter will now '3D-convolve' on each of the three channels of the (120 x 120 x 30) tensor.

**Methodology**

We have followed the approach of tuning the hyperparameters of our model to improve the accuracy and the overall usability. We have also modifying the model configurations such as dimension of the image fed into the network. Please find below the list of modifications which we have made for reaching at the optimum model –

1. cropping the images
2. resizing the images
3. number of images to be taken per video/sequence.
4. normalizing the images

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Experiment Number | Model | Model key configs and parameters | Result | Decision + Explanation |
| 1 | **Conv3D** | **Image Size: 100\*100**  **Batch Size: 10**  **Total params: 651,109**  **Trainable params: 650,997**  **Non-trainable params: 112** | **Training Accuracy: 0.38**  **Validation Accuracy: 0.29**  **Model File Size: 7.9 MB** | **Model underfits.**  **Model accuracy not up to the mark and we need to try tuning. We will not consider this model.** |
| 2 | **Conv3D** | **Image Size: 120\*120**  **Batch Size: 20**  **Total params: 1,117,061**  **Trainable params: 1,116,325**  **Non-trainable params: 736** | **Training Accuracy: 0.92**  **Validation Accuracy: 0.79**  **Model File Size: 675 MB** | **Model overfits hence model is not considered.** |
| Final Model | **Conv3D** | **Image Size: 120\*120**  **Batch Size: 10**  **Total params: 864,101**  **Trainable params: 863,989**  **Non-trainable params: 112**  **\*\*Normalized Data** | **Training Accuracy: 0.80**  **Validation Accuracy: 0.69**  **Model File Size: 10 MB** | **Model accuracy is satisfactory with great performance and hence we are selecting it.** |