**Gesture Recognition**

**Problem Statement:**

As a data scientist at a home electronics company which manufactures state of the art smart televisions. We want to develop a cool feature in the smart-TV that can recognise five different gestures performed by the user which will help users control the TV without using a remote.

Thumbs up : Increase the volume.

Thumbs down : Decrease the volume.

Left swipe : 'Jump' backwards 10 seconds.

Right swipe : 'Jump' forward 10 seconds.

Stop : Pause the movie.

**Understanding the Dataset**

The training data consists of a few hundred videos categorized into one of the five classes. Each video (typically 2-3 seconds long) is divided into a sequence of 30 frames (images). These videos have been recorded by various people performing one of the five gestures in front of a webcam - similar to what the smart TV will use.

**Objective**

Our task is to train different models on the 'train' folder to predict the action performed in each sequence or video and which performs well on the 'val' folder as well. The final test folder for evaluation is withheld - final model's performance will be tested on the 'test' set.

**Approach**

1. 3D Convolutions

2. CNN + RNN Architecture

|  |  |  |  |
| --- | --- | --- | --- |
| Sl. No | Experiment Details | Observations | Inferences/ Rectification |
| 1 | * Model Type: 3D Convolution * Image size= (120,120) * Epochs =10 * Batch size= 32 * Number of images =20 * Learning Rate = 0.001 * Dropouts =’ No’ | * Training Accuracy: 0.88 * Validation Accuracy: 0.25   Model is producing good accuracy but its highly overfitting. | Add dropouts in between the layers to reduce overfit. |
| 2 | * Model Type: 3D Convolution * Image size= (120,120) * Epochs =10 * Batch size= 32 * Number of images =20 * Learning Rate = 0.001 * Dropouts =’ Yes’ | * Training Accuracy: 0.71 * Validation Accuracy: 0.16   Model is producing good accuracy. Validation accuracy improved marginally still overfitting. | Increase dropouts,  Change batch size and other parameters. To test for rectification. |
| 3 | * Model Type: 3D Convolution * Image size= (120,120) * Epochs =10 * Batch size= 40 * Number of images =20 * Learning Rate = 0.001 * Dropouts =’ Yes’ | * Training Accuracy: 0.72 * Validation Accuracy: 0.17   The validation accuracy and training accuracy are slightly better for batch size 40. | Batch size =40  Seems to perform better than batch size 32 without heavily compromising on training time. |
| 4 | * Model Type: 3D Convolution * Image size= (120,120) * Epochs =10 * Batch size= 40 * Number of images =12 * Learning Rate = 0.001 * Dropouts =’ Yes’ | * Training Accuracy: 0.66 * Validation Accuracy: 0.19   The performance has dropped for number of images =12. | Increase the number of images(frames) to 15 |
| 5 | * Model Type: 3D Convolution * Image size= (120,120) * Epochs =10 * Batch size= 40 * Number of images =15 * Learning Rate = 0.001 * Dropouts = "Yes" | * Training Accuracy: 0.66 * Validation Accuracy: 0.16   The performance has dropped for number of images =15 | Changing the number of images did not improve the model performance. We decided to proceed with number of images=12. |
| 6 | * Model Type: 3D Convolution * Image size= (80,80) * Epochs =10 * Batch size= 40 * Number of images =12 * Learning Rate = 0.001 * Dropouts = "Yes" | * Training Accuracy: 0.53 * Validation Accuracy: 0.17   Both training and validation accuracy has dropped. The model is overfitting. | Changing the image size to (100,100). |
| 7 | * Model Type: 3D Convolution * Image size= (100,100) * Epochs =10 * Batch size= 40 * Number of images =12 * Learning Rate = 0.001 * Dropouts = "Yes" | * Training Accuracy: 0.57 * Validation Accuracy: 0.21   The accuracy has improved. | Image size (120,120) gave best accuracy. We proceed further with it to experiment with learning rate. |
| 8 | * Model Type: 3D Convolution * Image size= (120,120) * Epochs =10 * Batch size= 40 * Number of images =12 * Learning Rate = 0.01 * Dropouts = "Yes" | * Training Accuracy: 0.88 * Validation Accuracy: 0.54   The accuracy has improved significantly.  Overfitting has reduced. | Increasing the learning rate has improved the model performance and reduced the loss.  We retain this model and run 20 epochs of it. |
| 9 | * Model Type: 3D Convolution * Image size= (120,120) * Epochs =20 * Batch size= 40 * Number of images =12 * Learning Rate = 0.01 * Dropouts = "Yes" | * Training Accuracy: 0.89 * Validation Accuracy: 0.45   The accuracy has improved marginally but validation accuracy has reduced.  Model is overfitting but its very stable. | **BEST 3D convolution MODEL** |
| 10 | * Model Type: Convolution+Recurrent * Image size= (120,120) * Epochs =15 * Batch size= 40 * Number of images =12 * Learning Rate = 0.01 * Dropouts = "Yes" | * Training Accuracy: 0.98 * Validation Accuracy: 0.63   The accuracy achieved by combination of convolution and RNN is better. | Due to the GPU constraints we trained only for 15 epochs. More number of epochs would give better accuracy.  **BEST MODEL** |

**Based on the experiments conducted above combination of CNN and RNN gave the best training and validation accuracy. However we have to train the model on a larger sample and more epochs to obtain ideal results. The metrics used to evaluate has primarily validation accuracy followed by training accuracy (to detect overfitting)**

**PLEASE NOTE: H5 file has been uploaded at** [**https://drive.google.com/file/d/10W4mSw4ZGMcUEZxvxVzSQr5QzEj0YbHn/view?usp=sharing**](https://drive.google.com/file/d/10W4mSw4ZGMcUEZxvxVzSQr5QzEj0YbHn/view?usp=sharing)