**Gesture Recognition**

**Problem Statement**

Imagine you are working as a data scientist at a home electronics company which manufactures state of the art smart televisions. You want to develop a cool feature in the smart-TV that can recognize five different gestures performed by the user which will help users control the TV without using a remote. The gestures are continuously monitored by the webcam mounted on the TV. Each gesture corresponds to a specific command:

* 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

Each video is a sequence of 30 frames (or images)

**About 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. The data contains a 'train' and a 'val' folder with two CSV files for the two folders.

**Findings from different experiments are mentioned below.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Experiment Number** | **Model** | **Result** | **Decision + Explanation** |
| **1** | Conv3D  Batch Size=32  Image Size= 100\*100  Total number of frames taken are 20 | Total Parameters: 898,949  Training Accuracy:99%  Validation Accuracy:87% | Performance across every epoch has improved.  The validation accuracy is also increasing with every epoch. But the difference between accuracies is high. Kind of overfitting |
| **2** | Conv3D  Added Dropouts  Batch Size=32  Image Size= 100\*100 | Total Parameters: 898,949  Training Accuracy:98%  Validation Accuracy:87% | Even after adding dropouts there is no much change compared to above model, Still looks like overfitting |
| **3** | Conv3D  Added Dropouts and Batch Normalization  Batch Size=32  Image Size= 100\*100 | Total Parameters: 899,845  Training Accuracy:89%  Validation Accuracy:12% | The validation accuracy is not increasing and staying within 20% even though the train accuracy is increasing every epochs. Seems it's not getting trained properly. |
| **4** | Conv3D  Batch Size=64  Image Size= 100\*100 | Total Parameters: 898,949  Training Accuracy:85%  Validation Accuracy:75% | Even with batch size 64 there is no much change compared to above models for image size 100 |
| **5** | Conv3D  Batch Size=64  Image Size= 120\*120 | Total Parameters: 1,111,941  Training Accuracy:54%  Validation Accuracy:52% | By increasing the image size to 120 training accuracy is getting dropped and model becomes underfitting. |
| **6** | Conv3D  Batch Size=64  Image Size= 80\*80 | Total Parameters: 718,725  Training Accuracy:87%  Validation Accuracy:62% | By reducing the image size to 80, Performance across every epoch has improved.  The validation accuracy is also increasing with every epoch. However, the model is overfitting. |
| **7** | Conv3D  Batch Size=64  Image Size= 80\*80  In order to reduce parameters, lower the dense neurons to 64 from 128  Total number of frames passed are 16 by taking alternate ones | Total Parameters: 501,189  Training Accuracy:86%  Validation Accuracy:75% | Number of training parameters are reduced but still the difference between accuracies is high |
| **8(Best Model)** | **Conv3D**  **Batch Size=32**  **Image Size= 80\*80** | **Total Parameters: 501,189**  **Training Accuracy:93%**  **Validation Accuracy:93%** | **This looks the best model with good accuracy and less number of parameters** |
| **9** | Conv3D  Batch Size=32  Image Size= 100\*100 | Total Parameters: 591,301  Training Accuracy:97%  Validation Accuracy:81% | Parameters are increased and model is overfitting |
| **10** | Conv3D (Same as Model8 best till now)  Increasing the batch size to 128  Batch Size=128  Image Size= 80\*80 | Total Parameters: 501,189  Training Accuracy:62%  Validation Accuracy:61% | Overall Accuracy is low. |
| **11** | Conv3D (Same as Model8 best till now)  Increasing the batch size to 256  Batch Size=128  Image Size= 80\*80 | Total Parameters: 501,189  Training Accuracy:40 %  Validation Accuracy:39% | Overall Accuracy is low. Underfitting |
| **12** | Conv3D  Batch size 512 | Total Parameters: 501,189  OOM Error | Maintaining batch size 32 as we are getting good accuracy with lower number of parameters |
| **13** | Conv3D(Same as Model8)  Batch Size=32  Image Size= 80\*80  Changing the filter size to 2 to see if there is any change with lower number of parameters | Total params: 295,989  Training Accuracy:97 %  Validation Accuracy:87% | Though the trainable parameters are less. difference between training and validation accuracy is high |
| **14** | conv2d + LSTM  Batch Size=32  Image Size= 80\*80 | Total Parameters:991,525  Training Accuracy:76%  Validation Accuracy:37% | Model is overfitting |
| **14** | conv2d+ GRU  Batch Size=32  Image Size= 80\*80 | Total Parameters: 844,325  Training Accuracy:73%  Validation Accuracy:68% | Model is overfitting. |
| **15** | conv2d + LSTM  Batch Size=32  Image Size= 80\*80  Changed Dense neurons to 128 | Total Parameters: 1,000,101  Training Accuracy:96%  Validation Accuracy:62% | Parameters are increased and the difference between training and validation accuracy is high. |
| **16** | Transfer Learning MobileNet + GRU | Total Parameters: 3,438,469  Trainable params: 443,781  Training Accuracy:100%  Validation Accuracy:68% | Model is overfitting and trainable parameters are high in number |

**Best Model (Model8)**

Architecture - Conv3D and MaxPooling3D

Batch Size - 32

Image Size - 80

Filter size -3,3,3

Frames passed - 16 (Alternate frames)

Trainable Parameters - 501,189

Training Accuracy - 93%

Validation Accuracy - 93%