Gesture Recognition

Contents

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

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 which are categorized into 5 classes. Each video is divided into 30 frames of sequence(images). The videos are recorded by different people performing one of the 5 gestures in front of a webcam. It's similar to what Smart TV will be using.

We have provided with a .zip file contains train and test folders with two csv for the given folders.

Model Findings:

CONV 3D Model						
S.No	**Model Name**	**Model Type**	**Hyper Parameters**	**Accuracy & loss**	**No of Parameters**	**Observation**
1	MODEL 1 model_init_	Conv3D	total_frames = 30 num_frames = 15 classes = 5 image_height =100 image_width = 100 batch size = 8 num_epochs=25	**Train acc**:0.33 **Train loss**:1.50 **Test acc** 0.50 **Test loss**: 1.43	**Total**: 317,669 **Trainable:**317,557 **Non-Trainable:**112	This is a sample model with basic hyper parameters. The model is not leaning good on train data set. So, we are decreasing the height and width of the image. Used **SGD optimizer** with **lr** = 0.001
2	MODEL 2 model_exp2_	Conv3D	total_frames = 30 num_frames = 15 classes = 5 image_height = 50 image_width = 50	**Train acc**:0.43 **Train loss**:1.30 **Test acc** 0.68 **Test loss**: 1.03	**Total**: 96,485 **Trainable:**96,373 **Non-Trainable:**112	Resized the image from 100 to 50 and having the other parameters as same as above. Used **SGD optimizer** with **lr** = 0.001

			batch size = 8 num_epochs=25			There is very slight increase in accuracy values compared to MODEL 1.
3	MODEL 3 model_exp3_	Conv3D	total_frames = 30 num_frames = 15 classes = 5 image_height = 50 image_width = 50 batch_size = 4 num_epochs=25	**Train acc**:0.52 **Train loss**:1.83 **Test acc** 0.58 **Test loss**: 1.01	**Total**: 96,485 **Trainable:**96,373 **Non-Trainable:**112	Decreased the **batch_size** from 8 to 4 and the hyper parameters are same as model 2. Used **Adam optimizer** with **lr** = 0.0002 Not much of improvement observed in the model.
4	MODEL 4 model_exp4_	Conv3D	total_frames = 30 num_frames = 10 classes = 5 image_height = 50 image_width = 50 batch_size = 4 num_epochs=25	**Train acc**:0.35 **Train loss**:1.51 **Test acc** 0.45 **Test loss**: 1.53	**Total**: 96,485 **Trainable:**96,373 **Non-Trainable:**112	Decreased the num_frames from 15 to 10. Used **Adam optimizer** with **lr** = 0.0002 & **Dropout** = 0.5 No improvement in the model, lets decrease the dropout rate
5	MODEL 5 model_exp5_	Conv3D	total_frames = 30 num_frames = 10 classes = 5 image_height = 50 image_width = 50 batch_size = 4 num_epochs=25	**Train acc**:0.73 **Train loss**:0.76 **Test acc** 0.65 **Test loss**: 1.03	**Total**: 96,485 **Trainable:**96,373 **Non-Trainable:**112	Considering the same hyper parameters as MODEL 4 and decreased the **dropout rate** from 0.5 to 0.25. Observed, significant increase in accuracy for both train and test sets, With decrease in loss rate.

6	MODEL 6 model_exp6_	Conv3D	total_frames = 30 num_frames = 10 classes = 5 image_height = 50 image_width = 50 batch_size = 4 num_epochs=35	**Train acc**:0.86 **Train loss**:0.40 **Test acc** 0.80 **Test loss**: 0.63	**Total**: 96,485 **Trainable:**96,373 **Non-Trainable:**112	Here, Increased the epoch size from 25 to 35 by considering the same hyper parameters as MODEL 5. The model is performing quite well on both train and test sets with less trainable parameters. (**This is the final MODEL)**
7	MODEL 7 model_exp5_1_	Conv3D	total_frames = 30 num_frames = 12 classes = 5 image_height = 50 image_width = 50 batch_size = 4 num_epochs=25	**Train acc**:0.83 **Train loss**:0.42 **Test acc** 0.73 **Test loss**: 0.76	**Total**: 96,485 **Trainable:**96,373 **Non-Trainable:**112	It is an extension of MODEL 5 with increase in the num_frames to 12. To check the improvement of the model. There is a slight increase when compared to MODEL 5, but not compared MODEL 6
8	MODEL 8 model_exp6_1_	Conv3D	total_frames = 30 num_frames = 12 classes = 5 image_height = 50 image_width = 50 batch_size = 6 num_epochs=35	**Train acc**:0.78 **Train loss**:0.63 **Test acc** 0.65 **Test loss**: 0.82	**Total**: 96,485 **Trainable:**96,373 **Non-Trainable:**112	It is an extension to MODEL 6 with increase in both num_frames to 12 and batch_size to 6. This is not as good as MODEL 6.

CONV 2D Model - LSTM and GRU

9	MODEL 9	Conv2D, LSTM	total_frames = 30 num_frames = 10	**Train acc:**0.23	**Total**: 193,109	

	model_exp7_		classes = 5 image_height = 50 image_width = 50 batch_size = 4 num_epochs=35	**Train loss:**1.61 **Test acc** 0.25 **Test loss:** 1.59	**Trainable:**193,109	Considered, same hyper parameters as MODEL 5 and MODEL 6. But built Conv2D models using LSTM and GRU. • Total no. of parameters got increased compared to Conv3D. • Conv2D is not performing well on scores as compared to Conv 3D
10	MODEL 10 model_exp8_	Conv2D, GRU	total_frames = 30 num_frames = 10 classes = 5 image_height = 50 image_width = 50 batch_size = 4 num_epochs=35	**Train acc:**0.23 **Train loss:**1.61 **Test acc** 0.29 **Test loss:** 1.58	**Total**: 152,277 **Trainable:**152,777	
11	MODEL 11 model_exp9__	Conv2D, LSTM	total_frames = 30 num_frames = 12 classes = 5 image_height = 50 image_width = 50 batch_size = 4 num_epochs=25	**Train acc:**0.198 **Train loss:**1.64 **Test acc** 0.217 **Test loss:** 1.61	**Total**: 193,109 **Trainable:**193,109	
12	MODEL 12 model_exp10_	Conv2D, GRU	total_frames = 30 num_frames = 12 classes = 5 image_height = 50 image_width = 50 batch_size = 4 num_epochs=25	**Train acc:**0.26 **Train loss**:1.58 **Test acc** 0.38 **Test loss:** 1.53	**Total**: 152,277 **Trainable:**152,277	

Conclusion:

Based on the results when compared to Conv3D & Conv2D. Conv3D performing well on the dataset provided for Gesture Recognition.

Selected model: MODEL 6: Conv3D

Training Accuracy: 0.863

Validation Accuracy: 0.800

Batch Size: 4

Frames: 10

Image Height: 50

Image Width: 50

Drop out: Yes, 0.25

Batch Normalization: Yes

Layers: 8>> 16>> 32

Model File Name: model-00030-0.40803-0.86345-0.63196-0.80000.h5

model-00030-0.4080
3-0.86345-0.63196-0.8

- **Himaja Sri K**
- **A Anurag**