

Deep Learning Course Project - 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 recognise 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:

Gesture	Corresponding Action
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).

The image resolution, batch size and the number of frames were select after experiencing with 2 epochs.

So the base model was selected based on these factors.

- As we see from the above experiments "image resolution" and number of frames in sequence have more impact on training time than batch_size
- We can consider the Batch Size around 15-40
- We will change the resoulution 160 * 160, 120 * 120 according the model performance

SI NO.	Model	Parameters	Result	Decision +Explanation
1	Conv3D	1. Batch Size = 40 2. No. of Epochs = 20 3. image_height=160,image_widt h=160 4. frames_to_sample=20 5. Total parameters = 1,117,061 6. Trainable params: 1,116,325	Train_accuracy = 95.93 Val_accuracy = 26%	Base Model The val_accuracy was not improving. The moel was early stopped at 17 epochs Poor val_accuracy
2	Conv 3D with drop-out layer	7. Batch Size = 20 8. No. of Epochs = 25 9. image_height=160,image_widt h=160 10. frames_to_sample=20 11. Total parameters = 3,638,981 12. Trainable params: 3,637,477	Train_accuracy = 88.84 Val_accuracy = 80% val_loss: 0.3487	Thi is one o the best models with better val accuracy in Conv3D models. Better accuracy an be achieved with number o layers, augmentation techniques and changing the trainable parameters
3	Conv 3D With filter size Reduced to (2,2,2)	Reduce filter size to (2,2,2) and image res to 120 x 120, - Batch Size = 30 and No. of Epochs = 25	Train_accuracy = 68% Val_accuracy = 28% val_loss: 7.3240	Early stoped at epoch 23 Model totally overfitting with very poor val accuracy, which was not improving
4	Conv 3D with reduced number of parameters 3 hidden layers	filtersize=(3,3,3), and (2, 2, 2) in hidden layers dropout=0.25 image res to 120 x 120 Batch Size = 30 and No. of Epochs = 25 Total params: 504,709 Trainable params: 503,973	Categorical_acc uracy = 73.76% val_categorical_accuracy: 0.2500 val_loss: 3.7615	low memory foot print model the best validation accuracy is 0.25. which is not improing so the model was early stopped. Early stopped at 11th epoch as val_accuracy did not improve

5	CNN-LSTM Model	Total params: 1,657,445 Trainable params: 1,656,453 lstm_cells=64 image res to 120 x 120 Batch Size = 20 and No. of Epochs = 20 frames_to_sample=18 lstm_cells=128	categorical_accuracy: 95.55% val_categorical_accuracy: 75%	CNN - LSTM model we get a best validation accuracy of 75% Though can be better and definitely not the accuracy that we want to lock on.
6	With Augmentation	image_height=160,image_width=160 COLOR_BGR2GRAY Rotation And image cropped (3,3,3) Filter & 160x160 Image resolution - similar to Model 2 dense_neurons=256 Total params: 3,638,981 Trainable params: 3,637,477	categorical_accuracy: 79.79 val_categorical_accuracy: 65.00%	As we see more cases of overfitting, and the val accuracy also is not great
7	MobileNet Transfer Learning with GRU	image_height=120,image_width=120 frames_to_sample=16,batch_size=5,num_epochs=20 lstm_cells=128,dense_neurons=128,dropout=0.25 Total params: 3,840,453 Trainable params: 609,541	categorical_accuracy: 0.9811 - val_loss: 0.6252 - val_categorical_accuracy: 0.8400	We are not training the mobilenet weights and we see validation accuracy can be improved better than 84% at val_loss 0.6252. Let's train them as well and observe if there is performance improvement
8	Transfer Learning with GRU and training all weights with augmentation	image_height=120,image_width=120 frames_to_sample=16,batch_size=5,num_epochs=20 gru_cells=128,dense_neurons=128,dropout=0.25 Total params: 3,693,253 Trainable params: 3,669,317	categorical_accuracy: 98.42% - val_loss: 0.1024 - val_categorical_accuracy: 98.00% lr: 4.0000e-05	Awesome results! 98.4% Training accuracy and 98% validation accuracy With training time close to 48 mins.

After doing all the experiments, I finalized Model 8- GRU with Transfer learning MobileNet with all weights trained, which performed well.¶

Reason:

- Training Accuracy : 98.42%, Validation Accuracy : 98%
- Number of trainable Parameters 3,669,317
- Learning rate gradually decreasing after 8th Epoch

The best weights of GRU-CNNRNN transfer learning model:
model-00017-0.05718-0.98416-0.10244-0.98000.h5 (42.4 MB). I considered this weight for
model testing, -Training completed in about 53 minutes