CASE STUDY ON GESTURE RECOGNITION

Team Members:

Alok Narayan (Email ID: aloknarayan26@gmail.com)

Nandha Kishore (Email ID: snkishore007@gmail.com)

Problem Statement:

As a team of data scientist working for an electronics company manufacturing sophisticated state of the art televisions, we are asked to develop a new feature wherein the television would react and take action based on our hand gestures performed by us. This will help in controlling the TV without any remote and just based on our hand gestures. A webcam would capture the gestures wherein each hand gesture would correspond to a specific command.

GESTURE CORRESPONDING ACTION

1. Thumbs Up	-	Volume increase
2. Thumbs Down	-	Volume decrease
3. Left swipe	-	Rewind 10 seconds
4. Right Swipe	-	Fast forward 10 seconds
5. Stop	-	Pause

Objectives:

1. Generator:

The generator will take a batch of videos as input without any error. As different images of the videos are of different sizes, we will need to resize them and crop them. Additionally, we will also normalize them properly for effective modeling.

2. Model:

We need to develop a model keeping the principle of parsimony in mind. We need an effective trade off between accuracy and model complexity. Accuracy would be measured in terms of validation loss and model complexity in terms of number of parameters. The model must not be an underfit, neither should it over fit the training data.

3. Write Up:

This write up would contain the detailed procedure that we would follow in choosing the final model. The write up should start with the reason for choosing the base model, then highlight the reasons and metrics taken into consideration to modify and experiment to arrive at the final model.

MODELS

Sample Runs

We are taking 3 models

1. Batch size of 30 with 3 Epochs and frames to sample is 16.
2. Batch size of 40 with 3 Epochs and frames to sample is 16.
3. Batch size of 20 with 3 Epochs and frames to sample is 16.

We also made the following observations in this model

1. Categorical accuracy 67.25%
2. Validation accuracy 29%

Here, we can observe the above experiments image resolution and number of frames in sequence have more impact on training time than batch_size.

MODEL-A

CONV-3D model

We will take a batch size of 40 and 15 Epochs. Image size would be 120x120. We will also try to include the dropout of 25% to address over fitting of model and increase the epochs for better results.

We also made the following observations in this model:

1. No of Parameters 699,269
2. Categorical accuracy 95%
3. Validation accuracy 30%

We can clearly see model is overfitting.

MODEL-B

We are reducing the image size to 100×100 in this model.

Batch size = 40

Epoch = 15

Image size = 100×100

We will try a drop out of 25%.

We also made the following observations in this model:

1. No of Parameters 592,773

2. Categorical accuracy 94%

3. Validation accuracy 22%

No significant improvement, still overfitting

MODEL-C

Changing the Dense Neuron Parameter to 128 to see the performance.

Batch size = 30

Epoch = 15

Image size = 100×100

Let the dropout remain the same at 25%

We also made the following observations in this model:

1. No of Parameters 900,933

2. Categorical accuracy 87%

3. Validation accuracy 24%

Still no improvement, overfittinig exits

MODEL D - CNN- LSTM Model

We are introducing LSTM in this model.

Batch size = 20

Epoch = 20

Image size = 120X120

Let's keep drop out ratio also at 25% only.

We also made the following observations in this model:

1. No of Parameters 1,657,445

2. Categorical accuracy 95%

3. Validation accuracy 72%

Again, though both accuracies have increased, they are still not satisfactorily good.

MODEL E - CNN- GRU Model

We are using CNN + GRU in this model

Batch size = 20

Epoch = 20

lstm_cells = 128, dense neurons = 128

Image size = 120 * 120 and Dropouts = 25 %

Here is a plot showing training and validation accuracy

We also made the following observations in this model:

1. No of Parameters 2573925

2. Categorical accuracy 96%

3. Validation accuracy 78%

Again here, we haven’t got the desired results. But the performance has increased.

MODEL F - TRANSFER LEARNING and LSTM

Batch size = 5

Epoch = 20

lstm_cells = 128, dense neurons = 128

Image size = 120 X 120 and Dropouts = 25%

We also made the following observations in this model:

1. No of Parameters 3840453

2. Categorical accuracy 99%

3. Validation accuracy 74%

The loss of training has reduced compared to LSTM model , but validation loss has increased.

Model G: Transfer Learning with GRU

Batch size = 5

Epoch = 20

lstm_cells = 128, dense neurons = 128

Image size = 120 x 120 and Dropouts = 25 %

We also made the following observations in this model:

1. No of Parameters 3,693,253

2. Categorical accuracy 99%

3. Validation accuracy 92%

This is indeed the Best model and a near perfect model.

MODEL SUMMARY TABLE

Experiment Number	Model	Result	Decision + Explanation
1	Conv3D	Very Poor Train and validation accuracy	Reduce the size of image and check.
2	Conv3D	Veru poor train and validation accuracy still. Overfitting is observed.	Changing the dense neuron parameter to check the performance.
3	Conv3D	Still no improvement. Overfitting exists.	We are trying CNN + LSTM model.
4	CNN + LSTM	Model is performing better but could still improve.	Trying to use GRU instead of LSTM.
5	CNN + GRU	Accuracy has increased significantly and loss has been reduced.	Planning to use transfer learning methods to obtain better accuracy.
6	Transfer learning(MobileNet) + LSTM	Performance has decreased compared to CNN+ GRU.	Planning to use Transfer Learning with GRU.

7	**Transfer learning(MobileNet) + LSTM**	**This is the best model with highest Test and Validation Accuracy and Minimal loss.**	