Gesture Recognition Case Study Write-Up

**Problem Statement:**

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

**Experiments/Output:**

We will go ahead with Conv3D architecture instead of CNN+LSTM as the number of parameters would be 4 times in case of CNN +LSTM or 3 times in case of CNN+ GRU in comparison to a Conv3D architecture.

3D convolutions are a natural extension to the 2D convolutions. Just like in 2D conv, you move the filter in two directions (x and y), in 3D conv, you move the filter in three directions (x, y and z). In this case, the input to a 3D conv is a video (which is a sequence of 30 RGB images).

**Hyperparameter Tuning**

Please find below hyper tuning parameters that we experimented with:

* Batch sizes: 30, 40, 50, 64
* Epochs: 10, 20, 30
* Activation functions: RELU and ELU
* Optimizers: SGD, Adam
* Learning rates: .001, .0001, .005
* Network Size: 3-layer Conv3D and 4-layer Conv3D
* Sequence length: 12,15,18
* Image size: 120\*160, 120\*120, 100\*100, 90\*90

The final model we arrived at was based on the factors like;

1. Less training times.
2. Lowest possible parameters which helps make this model less resource intense
3. Better suited for a TV based applications

**Best Model parameters**

* Batch sizes: 30
* Epochs: 30
* Activation functions: RELU
* Optimizers: SGD
* Learning rates: .001
* Network Size: 4-layer Conv3D
* Sequence length: 18
* Image size: 90\*90
* Total parameters: 2,806,149
* Total time: 11.12 min
* Validation accuracy: 76%

**Experiment Details:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Experiment Number** | **Model** | **Action** | **Result + Explanation** | **Parameters** | **Validation accuracy** | **Training accuracy** | **Training time (mins)** |
| 1 | Conv3D | Need to ensure that all the inputs are of same size | Extract shape of each image and find unique ones i.e. Identified images with sizes - (120, 160, 3), (360, 360, 3) | NA | NA | NA | NA |
| 2 | Conv3D | Model with 10 epochs and 40 batch size on nimble box, train data = 350 | Throwing dead kernel errors on nimble box | NA | NA | NA | NA |
| 3 | Conv3D | Model with 10 epochs and 30 batch size, train data=350 and **rewrote the generator function** | Model completed all epochs -**Model was overfitting due to less data**  - LR=.001 - SGD optimizer - 3-layer Conv3D, (32,64,128), Dense -128,5 - relu activation - batch optimization - dropout | 8,948,101 | 61% | 76% | 3.59 |
| 4 | Conv3D | Model with 10 epochs, 40 batch size, train data=663 | -**Model was overfitting with full data as well**  - LR=.001 - SGD optimizer, momentum =0.5, nesterov=True - 3 layer Conv3D, (32,64,128) , Dense -128,5 - relu activation - batch optimization - dropout - after 3rd layer | 8,948,101 | 62 | 77.78 | 7.17 |
| 5 | Conv3D | Model with 10 epochs, 50 batch size, train data=663 | **OOM issues encountered and decided to go with 40 batch size** | 8,948,101 | NA | NA | NA |
| 6 | Conv3D | Model with 10 epochs, **40 batch size**, train data=663 ,**adding dropout after 2nd layer** | -**Model is no more overfitting after adding another dropout** - LR=.001 - SGD optimizer, momentum =0.5, nesterov=True - 3 layer Conv3D , Dense -128,5 - relu activation - batch optimization - dropout - after 2nd , 3rd layer | 8,948,101 | 71 | 69.78 | 6.58 |
| 7 | Conv3D | Model with **20 epochs** , 40 batch size, train data=663 , **used ELU activation** | **ELU activation didn’t worked as good as RELU , decided to proceed with RELU activation** - LR=.001 - SGD optimizer, momentum =0.5, nesterov=True - 3 layer Conv3D (32,64,128), Dense -128,5 - elu activation - batch optimization - dropout - after 2nd , 3rd layer | 8,948,101 | 69 | 74.39 | 12.39 |
| 8 | Conv3D | Model with 20 epochs, 40 batch size, train data=663, **Adam optimizer** | **Model didn't learn using ADAM. We finalize SGD** -LR=.001 - Adam optimizer - 3 layer Conv3D (3\*3\*3 filter) - (32,64,128), Dense -128,5 - relu activation - batch optimization - dropout - after 2nd , 3rd layer | 8,948,101 | 30 | 30.79 | 6.07 |
| 9 | Conv3D | Model with 20 epochs, 40 batch size, train data=663, **Increase no. of neurons in Conv3D to (64,128,256)** | **Threw OOM error**  -LR=.001 - SGD optimizer, momentum =0.5, nesterov=True - 3 layer Conv3D (3\*3\*3 filter) - (64,128,256) , Dense -128,5 - relu activation - batch optimization - dropout - after 2nd , 3rd layer | NA | NA | NA | NA |
| 10 | Conv3D | Model with 20 epochs, **30 batch size,** train data=663, **Increase no. of neurons in Conv3D to (64,128,256)** | **Threw OOM error**  -LR=.001 - SGD optimizer, momentum =0.5, nesterov=True - 3 layer Conv3D (3\*3\*3 filter) - (64,128,256) , Dense -128,5 - relu activation - batch optimization - dropout - after 2nd , 3rd layer | NA | NA | NA | NA |
| 11 | Conv3D | Model with 20 epochs, 40 batch size, train data=663, **4 CONV layers - (2\*2\*2 filter)** | **Threw OOM error**  LR=.001 - SGD optimizer, momentum =0.5, nesterov=True - 4 layer Conv3D (2\*2\*2 filter) - relu activation - batch optimization - dropout - after 2nd , 3rd, 4th layer | NA | NA | NA | NA |
| 12 | Conv3D | Model with 20 epochs, 40 batch size, train data=663, 3 CONV layers - (3\*3\*3 filter), **removed dropout from dense layer** | -Model is heavily overfitting and giving 65% validation accuracy vs 99 training accuracy -This is good as we want CNN to overfit on training data -Early stopped after 13 epochs - LR=.001 - SGD optimizer, momentum =0.5, nesterov=True - 3 layer Conv3D (32,64,128) , Dense -128,5 - relu activation - batch optimization - dropout - after 2nd , 3rd layer | 8,948,101 | 65 | 99 | 9.12 |
| 13 | Conv3D | Model with 20 epochs, 40 batch size, train data=663, 3 CONV layers - (3\*3\*3 filter), **added drop out at dense layer with 0.25** | -Model is still overfitting and giving 67% validation accuracy vs 86 training accuracy -Adding dropout at dense layer reduced the overfitting by 14% -Early stopped after 19 epochs - LR=.001 - SGD optimizer, momentum =0.5, nesterov=True - 3 layer Conv3D (32,64,128) , Dense -128,5 - relu activation - batch optimization - dropout - after 2nd (0.25), 3rd layer (0.25), dense layer (0.25) | 8,948,101 | 67 | 86.14 | 13.48 |
| 14 | Conv3D | Model with 20 epochs, 40 batch size, train data=663 , 3 CONV layers - (3\*3\*3 filter), **added drop out at 1st layer 0.25** | -Model is still overfitting and giving 48% validation accuracy vs 67 training accuracy -Early stopped after 11 epochs - LR=.001 - SGD optimizer, momentum =0.5, nesterov=True - 3 layer Conv3D (32,64,128) , Dense -128,5 - relu activation - batch optimization - dropout - after 1st(0.25), 2nd(0.25), 3rd layer (0.25), dense layer (0.25) | 8,948,101 | 48 | 66.99 | 7.43 |
| 15 | Conv3D | Model with 20 epochs, 40 batch size, train data=663 , 3 CONV layers - (3\*3\*3 filter) , **drop out at dense layer changes to 0.5** | -Model is no more overfitting and giving 75% validation accuracy  **-One of the best model in all iterations we identified** - LR=.001 - SGD optimizer, momentum =0.5, nesterov=True - 3 layer Conv3D (32,64,128) , Dense -128,5 - relu activation - batch optimization  dropout - after 2nd(0.25), 3rd layer (0.25), dense layer (0.5) | 8,948,101 | 75 | 80.55 | 12.27 |
| 16 | Conv3D | Model with **30 epochs**, 40 batch size, train data=663, 3 CONV layers - (3\*3\*3 filter), **LR-.0005** | -Model is overfitting and giving 69% validation accuracy - LR=.0005 - SGD optimizer, momentum =0.5, nesterov=True - 3 layer Conv3D (32,64,128) , Dense -128,5 - relu activation - batch optimization  dropout - after 2nd(0.25), 3rd layer (0.25), dense layer (0.5) | 8,948,101 | 69 | 83.1 | 13.3 |
| 17 | Conv3D | Model with **20 epochs**, **30 batch size**, train data=663, 3 CONV layers - (3\*3\*3 filter), **Seq=18** | -Model is no more overfitting and giving 73% validation accuracy - LR=.001 -Seq length = 18 - SGD optimizer, momentum =0.5, nesterov=True - 3 layer Conv3D (32,64,128) , Dense -128,5 - relu activation - batch optimization  dropout - after 2nd(0.25), 3rd layer (0.25), dense layer (0.5) | 17,615,237 | 73 | 76.25 | 19.23 |
| 18 | Conv3D | Model with **20 epochs**, **30 batch size**, train data=663 , 3 CONV layers - (3\*3\*3 filter), **Seq=18, image size=90\*90** | -Model is underfitting and overfits later with more epochs and giving 67% validation accuracy - LR=.001 -Seq length = 18 - SGD optimizer, momentum =0.5, nesterov=True - 3 layer Conv3D (32,64,128) , Dense -128,5 - relu activation - batch optimization  dropout - after 2nd(0.25), 3rd layer (0.25), dense layer (0.5) | 13,388,165 | 67 | 66.52 | 12.26 |
| 19 | Conv3D | Model with **30 epochs**, **30 batch size**, train data=663, **4 CONV layers - (3\*3\*3 filter)**, **Seq=18, image size=90\*90,** | -Model is no more overfitting and giving 75% validation accuracy - LR=.001 -Seq length = 18 - SGD optimizer, momentum =0.5, nesterov=True - 3 layer Conv3D (64,128,256,256) , Dense -256,5 - relu activation - batch optimization  dropout - after 2nd(0.25), 3rd layer (0.25),4th layer (0.25), dense layer (0.5) | 4,524,037 | 73 | 74.65 | 16.11 |
| 20 | Conv3D | Model with 20 epochs, 40 batch size, train data=663 , 3 CONV layers - (3\*3\*3 filter) | -Model is still overfitting and giving 69% validation accuracy vs 84 training accuracy -Early stopped after 17 epochs - LR=.001 - SGD optimizer, momentum =0.5, nesterov=True - 3 layer Conv3D (32,64,128) , Dense -128,5 - relu activation - batch optimization - **dropout - after 2nd , 3rd and dense layer** | 8,948,101 | 69 | 84 | 12 |
| 21 | Conv3D | Model with 20 epochs , 40 batch size, train data=663 , 3 CONV layers - (3\*3\*3 filter) ,**L2 Kernel regularization on layers** | -Model is not overfitting and giving 75% validation accuracy vs 80% training accuracy **-Second best model in all iterations** - LR=.001 - SGD optimizer, momentum =0.5, nesterov=True - 3 layer Conv3D (32,64,128) , Dense -128,5 - relu activation - batch optimization - dropout - after 2nd , 3rd and dense layer | 8,948,101 | 75 | 80 | 14 |
| 22 | Conv3D | Model with 20 epochs, 40 batch size, train data=663, 3 CONV layers - (3\*3\*3 filter) ,L2 Kernel and **Bias regularization** | -Model has not improved much with addition of bias regularization and giving 68% validation accuracy vs 79% training accuracy -Early stopped after 20 epochs - LR=.001 - SGD optimizer, momentum =0.5, nesterov=True - 3 layer Conv3D (32,64,128) , Dense -128,5 - relu activation - batch optimization - dropout - after 2nd , 3rd layer and dense layer | 8,948,101 | 68 | 79 | 14 |
| 23 | Conv3D | Model with 20 epochs, 40 batch size, train data=663 , 3 CONV layers - (3\*3\*3 filter) ,L2 Kernel regularization | -Model accuracy dropped but has same performance , 56% validation accuracy vs 56% training accuracy - **LR=.0001** - SGD optimizer, momentum =0.5, nesterov=True - 3 layer Conv3D (32,64,128) , Dense -128,5 - relu activation - batch optimization - dropout - after 2nd , 3rd and dense layer | 8,948,101 | 56 | 56 | 14 |
| **Final Model** | Conv3D | Model with **30 epochs**, **30 batch size**, train data=663, **4 CONV layers - (3\*3\*3 filter)**, **Seq=18, image size=90\*90 , L2 Kernel regularization** | **-Model is no more overfitting and giving 76% validation accuracy -This was the best CNN / combination of hyperparameters that we got and we decided to make this our final model** - LR=.001 -Seq length = 18 - SGD optimizer, momentum =0.5, nesterov=True - 3 layer Conv3D (32,64,128) , Dense -128,5 - relu activation - batch optimization  dropout - after 2nd(0.25), 3rd layer (0.25), dense layer (0.5) | 2,806,149 | 76 | 77.93 | 11.12 |