

Neural Networks Project - Gesture Recognition

Institute: IIIT, Bangalore & upGrad

Course: PG Diploma in ML & AI (Batch: C 12)

Submission Date: 17-AUG-2020

Submitted By:

1. Prannoy Prashad

2. Anjali Sharma

Problem Statement: We need 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.

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

Understanding the Dataset

The training data consists of a few hundred videos categorised into one of the five classes. Each video is a sequence of 30 frames (or 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.

Our task is to train a model on the train folder which performs equally well on validation folder. Final model's performance will be tested on the 'test' set.

EX No.	Model	Hyper-parameter	Result	Decision & Explanation
1	Conv3d	Batch Size: 60 No. of image frames: 30 Image Height: 160 Image Width: 160 Learning Rate: 0.0001 Optimiser: Adam	Categorical Accuracy - 60% Validation Categorical Accuracy - 57%	Though the model is not overfitting but accuracy is very low. So we need to fine tune our model.
2	Conv3d	Batch Size: 70 No. of image frames: 30 Image Height: 160 Image Width: 160 Learning Rate: 0.0001 Optimiser: Adam	OOM when allocating tensor with shape[70,8,30,160,160] and type float on /job:localhost/replica:0/task:0/device:GPU:0 by allocator GPU_0_bfc	We increased the batch size so that we can decrease the number of iterations in every epoch. But it results in OOM
3	Conv3d	Batch Size: 60 No. of image frames: 30 Image Height: 120 Image Width: 120 Learning Rate: 0.0001 Optimiser: Adam	Categorical Accuracy - 65% Validation Categorical Accuracy - 60%	This time we decreased the batch size as we got OOM on batch size 70. We also reduced the image dimensions. The accuracy increased a little but we need to fine tune it more.
4	Conv3d	Batch Size: 60 No. of image frames: 30 Image Height: 120 Image Width: 120 Learning Rate: 0.0001 Optimiser: SGD	Categorical Accuracy - 41% Validation Categorical Accuracy - 60%	We experimented with the optimiser. Instead of Adam optimiser, we used SGD optimiser but our accuracy further dropped. So we will switch back to Adam

5	Conv3d	Batch Size: 50 No. of image frames: 30 Image Height: 100 Image Width: 100 Learning Rate: 0.001 Optimiser: Adam	Categorical Accuracy - 90% Validation Categorical Accuracy - 83%	This model is quite acceptable as the accuracy has increased significantly after reducing the image dimensions.
6	Conv3d	Batch Size: 50 No. of image frames: 30 Image Height: 100 Image Width: 100 Learning Rate: 0.001 Optimiser: Adam img_idx in generator : range[1,30]	Categorical Accuracy - 92% Validation Categorical Accuracy - 83% Time taken to run the model increased with no significant increase in accuracy.	We used all 30 frames in generator function but the models were taking a lot of time to train and accuracy did not increase in that proportion.
7	Conv3d	img_idx in generator : range[1,30,2]	The model train time decreased and model was as good as before.	We are now using all odd numbered frames. We noticed that the time taken to train each model has reduced significantly.
8	Conv3d	Increasing epochs	Accuracy increased as we increased the number of epochs	We initially used no. of epochs = 10. But as we increased the epochs , we realised that we were getting better accuracy for epoch around 30. So for all our models we then started using no of epoch as 30.

9	Conv3d	Batch Size: 50 No. of image frames: 30 Image Height: 160 Image Width: 160 Learning Rate: 0.001 Optimiser: Adam Dropout of 0.5 after every layer	Categorical Accuracy - 49% Validation Categorical Accuracy - 16%	After adding dropout after every layer, though the model should not have over fit but it still is and the accuracy is also quite low. So we need to do further experiments with dropout.
10	Conv3d	No dropout Batch Size: 50 No. of image frames: 30 Image Height: 160 Image Width: 160 Learning Rate: 0.001 Optimiser: Adam	Categorical Accuracy - 87% Validation Categorical Accuracy - 75%	The model has improved a lot after we removed all dropouts, used a learning rate of 0.001 with Adam optimiser. Though we need to improve more as the model seems to over fit.
11	Conv3d	dropout of 0.5 in last 2 layers Batch Size: 30 No. of image frames: 30 Image Height: 160 Image Width: 160 Learning Rate: 0.001 Optimiser: Adam	Categorical Accuracy - 84% Validation Categorical Accuracy - 79%	In this model though the training accuracy has dipped a little but validation accuracy has increased, making the model less likely to over fit.

12	Conv3d	<p>dropout of 0.5 in last 2 layers</p> <p>Batch Size: 50</p> <p>No. of image frames: 30</p> <p>Image Height: 120</p> <p>Image Width: 120</p> <p>Learning Rate: 0.001</p> <p>Optimiser: Adam</p>	<p>Categorical Accuracy - 87%</p> <p>Validation Categorical Accuracy - 79%</p>	<p>Though the categorical accuracy has increased but validation accuracy has reduced. So we need to experiment more</p>
13	Conv3d	<p>Added another Conv2d layer</p> <p>dropout of 0.5 in last 2 layers</p> <p>Batch Size: 50</p> <p>No. of image frames: 30</p> <p>Image Height: 120</p> <p>Image Width: 120</p> <p>Learning Rate: 0.001</p> <p>Optimiser: Adam</p>	<p>Categorical Accuracy - 89%</p> <p>Validation Categorical Accuracy - 79 %</p>	<p>After adding another layer the no. of trainable parameters have increased drastically but accuracy hasn't . So we decided to drop this model</p>
14	Conv3d	<p>Kernel size 4*4 used</p> <p>dropout of 0.5 in last 2 layers</p> <p>Batch Size: 50</p> <p>No. of image frames: 30</p> <p>Image Height: 120</p> <p>Image Width: 120</p> <p>Learning Rate: 0.001</p> <p>Optimiser: Adam</p>	<p>Categorical Accuracy - 86%</p> <p>Validation Categorical Accuracy - 53 %</p>	<p>In all the above models we were using a kernel size of 3*3. After changing it to 4*4 the model started overfitting and validation accuracy dropped a lot.</p>

15	Conv2d +LSTM (Using Mobile net)	Batch Size: 50 No. of image frames: 30 Image Height: 120 Image Width: 120 Learning Rate: 0.001 Optimiser: Adam Dropout 0.25	Categorical Accuracy - 34% Validation Categorical Accuracy -21 %	Though LSTM has reduced no. of trainable parameters but its performing very poorly
16	Conv2d +GRU (Using Mobile net)	Batch Size: 50 No. of image frames: 30 Image Height: 120 Image Width: 120 Learning Rate: 0.001 Optimiser: Adam Dropout 0.25	Categorical Accuracy - 68% Validation Categorical Accuracy -22 %	Though it is performing better than LSTM but the model is clearly over fitting.

Final Model: The best model according to our analysis is **Conv3D+MaxPooling3D** where the hyper-parameters are

Batch Size: 50

No. of image frames: 30

Image Height: 100

Image Width: 100

Learning Rate: 0.001

Optimiser: Adam

Here we got the **training accuracy** as approx. **90%** and validation accuracy as **83%** where total number of trainable parameters are **2881589**.

