Gesture Recognition

upGrad Assignment

Batch C55

Syed Abdul Rahim

Rounak Rai

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

We intend to use Conv3D to solve this deep learning problem.

Model Findings:

#	Hyper Parameters	Params Count	Accuracy	Remarks	Action
1	**Batch Size:** 1 **Frames:** 10 **Image Height:** 50 **Image Width:** 50 **Drop out:** No	**Total:** 366661 (1.40 MB) **Trainable:** 366437 (1.40 MB) **Non-trainable:** 224 (896.00 Byte)	**Training:** 1.0 **Validation:** 0.86	**1.** Started with batch size 1 to check the model is working **2.** Validation accuracy is not bad	**1.** Increase Batch Size **2.** Increase Image Height and Width

				3. High difference between Training vs Validation, shows model to be overfitting	
2	**Batch Size:** 32 **Frames:** 10 **Image Height:** 100 **Image Width:** 100 **Drop out:** No **Batch Normalization:** Yes **Layers:** 16 > 32 > 64 >> 128 >> 5	**Total:** 1251397 (4.77 MB) **Trainable:** 1251173 (4.77 MB) **Non-trainable:** 224 (896.00 Byte)	**Training:** 1.0 **Validation:** 0.89	**1.** Increased batch size + Image size, might have degraded performance **2.** Validation Accuracy has increased **3.** Training accuracy at 100% and High difference between Training vs Validation, shows model to be overfitting	**1.** Introduced Dropout with 0.5
3	**Batch Size:** 32 **Frames:** 10 **Image Height:** 100 **Image Width:** 100 **Drop out:** Yes, 0.5 **Batch Normalization:** Yes **Layers:** 16 > 32 > 64 >> 128 >> 5	**Total:** 1251397 (4.77 MB) **Trainable:** 1251173 (4.77 MB) **Non-trainable:** 224 (896.00 Byte)	**Training:** 0.65 **Validation:** 0.77	**1.** Suffered in both Training and Validation Accuracy **2.** Training accuracy is more than Validation, issue in the learning itself	**1.** Change architecture starting with 32 layers
4	**Batch Size:** 32 **Frames:** 10 **Image Height:** 100 **Image Width:** 100 **Drop out:** Yes, 0.5 **Batch Normalization:** Yes **Layers:** 32 > 64 > 128 >> 256 >> 5	**Total:** 5000325 (19.07 MB) **Trainable:** 4999877 (19.07 MB) **Non-trainable:** 448 (1.75 KB)	**Training:** 0.89 **Validation:** 0.82	**1.** Training and Validation Accuracy increased **2.** Validation accuracy is not in acceptable range **3.** High number of trainable params increasing the model size	**1.** Increase batch size to 64 **2.** Crop image to 50 x 50

5	Batch Size: 64 Frames: 10 Image Height: 50 Image Width: 50 Drop out: Yes, 0.5 Batch Normalization: Yes Layers: 32 > 64 > 128 >> 256 >> 5	Total: 1461381 (5.57 MB) Trainable: 1460933 (5.57 MB) Non-trainable: 448 (1.75 KB)	Training: 0.79 Validation: 0.79	1. Impacted performance a lot 2. Validation accuracy didn't increase but rather Training accuracy decreased 3. High difference between Training vs Validation, shows model to be overfitting	1. Reduce batch size to 32 2. Change architecture to start with 8 layers
6	Batch Size: 32 Frames: 10 Image Height: 50 Image Width: 50 Drop out: Yes, 0.5 Batch Normalization: Yes Layers: 8 > 16 > 32 >> 64 >> 5	Total: 92325 (360.64 KB) Trainable: 92213 (360.21 KB) Non-trainable: 112 (448.00 Byte)	Training: 0.38 Validation: 0.51	1. Training and Validation Accuracies are least of all models, not acceptable	1. As we already have some good performing models, will choose one from those

Conclusion:

Selected the best model from Model – 4 with,

Training Accuracy: 0.89

Validation Accuracy: 0.82

Batch Size: 32

Frames: 10

Image Height: 100

Image Width: 100

Drop out: Yes, 0.5

Batch Normalization: Yes

Layers: 32 > 64 > 128 >> 256 >> 5

Model File Name: model-00012-0.27254-0.89347-0.77396-0.82031.h5

Model File Size: 57.2 MB

[End of Document]