Gesture Recognition Case Study

Shameer Sheik – Team Member, ML C56 batch

**Problem Statement**

As a data scientist at a home electronics company which manufactures state of the art smart televisions. We 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.

* 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.

The data is in a zip file. The zip file contains a 'train' and a 'val' folder with two CSV files for the two folders. These folders are in turn divided into subfolders where each subfolder represents a video of a particular gesture. Each subfolder, i.e. a video, contains 30 frames (or images). Note that all images in a particular video subfolder have the same dimensions but different videos may have different dimensions. Specifically, videos have two types of dimensions - either 360x360 or 120x160 (depending on the webcam used to record the videos). Hence, you will need to do some pre-processing to standardize the videos.

Each row of the CSV file represents one video and contains three main pieces of information - the name of the subfolder containing the 30 images of the video, the name of the gesture and the numeric label (between 0-4) of the video.

Task is to train a model on the 'train' folder which performs well on the 'val' folder as well (as usually done in ML projects). We have withheld the test folder for evaluation purposes - your final model's performance will be tested on the 'test' set.

**Pre-processing / Standardization:**

Resize all the selected different size images to same size 100 \*100, i.e., height 100, width 100.

Consider the number of epochs to be 25.

Batch size = 8.

# Experiment 1

### Model:

### Initial Check after image resize to 100\*100 and Optimizer as Adam.

* Number of frames: 15, Number of Epochs: 25
* Batch size = 8.
* Conv 3D model with 8, 16, 32 filters, Dense 64 nodes

### Result:

* **Training Accuracy:** 0.35%
* **Validation Accuracy:** 0.5%

### Plot:

### 

**Conclusion:**

From the graph we can see that the model is performing better on the validation data than on the training data. This indicates that the model is too simple and has fewer that required training parameters.

# Experiment 2

Model:

* Reducing the size of the image from 100 to 50, i.e., height 50, width 50
* Number of frames: 15, Number of Epochs: 25
* Batch size = 8.
* Conv 3D model with 8, 16, 32 filters, Dense 64 nodes

Result:

* **Training Accuracy:** 0.34%
* **Validation Accuracy:** 0.48%

**Plot:**

A graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of

Description automatically generated

**Conclusion:** Scaling to smaller images decreased both training accuracy and validation accuracy.

# Experiment 3

Model:

* Reduce the size of the image from 50 to 25, i.e., height 25, width 25.
* Number of frames: 15, Number of Epochs: 25
* Batch size = 8.
* Conv 3D model with 8, 16, 32 filters, Dense 64 nodes

Result:

* **Training Accuracy:** 0.63%
* **Validation Accuracy:** 0.63%

Plot:

A graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of

Description automatically generated

**Conclusion:** Scaling from 50 to 25 increased the training and validation accuracy

# Experiment 4

Model:

* Decreasing the batch size from 8 to 4, image size as 50\*50 i.e., height 50, width 50.
* Number of frames: 15, Number of Epochs: 25
* Batch size = 4.
* Conv 3D model with 8, 16, 32 filters, Dense 64 nodes

Result:

* **Training Accuracy:** 0.43%
* **Validation Accuracy:** 0.68%

**Plot:**

A graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of

Description automatically generated

**Conclusion:** Reducing the batch size has reduced the training accuracy but improved the validation accuracy. Indirectly, it reduced the overfitting.

# Experiment 5

Model:

* Changing the optimizer from Adam to SGD, image size as 50\*50 i.e., height 50, width 50.
* Number of frames: 15, Number of Epochs: 25
* Batch size = 4.
* Conv 3D model with 8, 16, 32 filters, Dense 64 nodes

Result:

* **Training Accuracy:** 0.53%
* **Validation Accuracy:** 0.69%

Plot:

A graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of

Description automatically generated

**Conclusion:** Validation accuracy increased by 1%, but training accuracy is improved, meaning model is prone to overfitting compared to Adam optimizer.

# Experiment 6

Model:

* Reducing the number of frames from 15 to 10, image size as 50\*50 i.e., height 50, width 50.
* Number of frames: 10, Number of Epochs: 25
* Batch size = 4.
* Conv 3D model with 8, 16, 32 filters, Dense 64 nodes

Result:

* **Training Accuracy:** 0.25%
* **Validation Accuracy:** 0.41%

Plot:

A graph of a line and a line graph

Description automatically generated with medium confidence

**Conclusion:** The training and validation accuracy has decreased after reducing the number of frames from 15 to 10

# Experiment 7

Model:

* Reducing the dropout rate from 0.5 to 0.25, image size as 50\*50 i.e., height 50, width 50.
* Number of frames: 10,
* Batch size = 4.
* Conv 3D model with 8, 16, 32 filters, Dense 64 nodes

Result:

* **Training Accuracy:** 0.62%
* **Validation Accuracy:** 0.7%

Plot:

A graph of a line and a line

Description automatically generated with medium confidence

**Conclusion:** A drop-out rate of 0.25 gives up a better training and validation accuracy

# Experiment 8 / Final Best Model

Model:

* Increasing the number of epochs from 25 to 35, image size as 50\*50, i.e., height 50, width 50.
* Number of frames: 10,
* Batch size = 4.
* Conv 3D model with 8, 16, 32 filters, Dense 64 nodes

Result:

* **Training Accuracy:** 0.9%
* **Validation Accuracy:** 0.81%

Plot:

A graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of

Description automatically generated

**Conclusion:**

After 35 epochs the training and validation accuracy changes to 90% and 81% respectively. At 30th Epoch, we got the best result with better accuracy in both training and validation, along with the minimum loss.