**Gesture Recognition Case Study**

* Sudalaikannu P Vaithiar
* Lochan Raj T D

**Problem Statement:**

As a data scientist at a home electronics company which manufactures state of the art smart televisions. We want develop a cool feature in the Smart TV that can recognize different gestures performed by the user which will users control the TV without using a remote.

1. Thumbs up : Increase the volume
2. Thumbs down : Decrease the volume
3. Left swipe : ‘Jump’ backward 10 seconds
4. Right swipe : ‘Jump’ forward 10 seconds
5. Stop : Pause the movie.

**Dataset:**

The training data consists of a few hundred videos categorized into one of the five classes. Each videos (typically 2-3 seconds long) is divided into a sequences of 30 frames (images). These videos have been recorded by various people performing one of the five gestures in front of webcam – to what the smart TV will use.

**Generator:**

Generator is the most important part of this code. In this code, we use generator to pre-process as we have 2 different dimensions as well as create video frames batch. Two generators are used in the code.

* generator: In this generator, source path, folder path, batch size are inputs which generator takes. These generator crops the images and resize them as the images are of 2 different shape. If we don’t then error will be thrown. Normalization of data is also part of pre-preprocess.
* generator\_ex: In this generator, it does same as generator but it also performs the part of converting the images to grayscale.

**Models and Experiments:**

In this code, there are number of models and experiments to come to the final models.

|  |  |  |  |
| --- | --- | --- | --- |
| Model and Experiment | Categorical Accuracy | Validation Accuracy | Changes made |
| Model A | 0.4179 | 0.5100 | generator was used. Optimizer Adam used with learning rate of 0.001 |
| Model B | 0.1990 | 0.2100 | generator\_ex was used without learning rate. |
| Model B Experiment 1 | 0.4378 | 0.4200 | Changed the number of frame, image height and image width |
| Model B Experiment 2 | 0.6965 | 0.4700 | Changing Batch size to 20 |
| Model B Experiment 3 | 0.8402 | 0.4950 | Changing Batch size to 40 |
| Model B Experiment 4 | 0.8385 | 0.4650 | Changing to Optimizer Adadelta |
| Model B Experiment 5 | 0.2660 | 0.3150 | Changing epoch to 20 |
| Model C Experiment 6 | 0.2239 | 0.2400 | Added more Dense layer |
| Model D Experiment 7 | 0.2189 | 0.2200 | Changed channel to 1, image height and image width and epoch to 30. |
| Model E Experiment 8 | 0.2289 | 0.2600 | Changed channel to 3, image height and image width and epoch to 10. |
| Model F Experiment 9 | 0.1891 | 0.3050 | Changed Batch size to 20 and epoch to 20 |
| Model G Experiment | 0.2239 | 0.2200 | Changed Batch size to 30 and epoch 10 |
| Final Model | 0.9789 | 0.7750 | After considering the previous models and Experiments, The models with optimizer changed Adam input size, Batch size, epoch and filters |

After the final model, we obtained the final .h5 after lasted epoch.

model\_init\_2021-11-1811\_52\_10.520291/model-00020-0.07526-0.97893-0.78052-0.77500.h5