## **Gesture Recognition using Deep learning**

By Sujay Kondanekar and Venkatesh H V

**Problem Statement**

As a data scientist at a home electronics company which manufactures state of the art smart televisions, I should develop a cool feature in the smart-TV that can recognise five different gestures performed by the user which will control the TV without using a remote.

**Training dataset**

The training data consists of a hundreds of videos categorized into one of the five classes. Each video (typically 2-3 seconds long) is divided into a sequence of 30 frames (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.

**Experiments performed**

Generator code - In the generator, the images are pre-processed to have images of 2 different dimensions and create a batch of video frames. The number of folders and videos is made divisible by the batch size number. The images are augmented to perform –

1) Edge enhancement

2) enhance image detailing

3) image sharpening and

4) Brightness enhancement

Once the generator code is executed the idea of the solution is to first try out model building using Conv3D architecture and vary batch size and number of epochs. Also, the model will be supplemented with more augmentation to reduce dimensions of the feature maps using Maxpooling. We will also experiment models using Conv2D + GRU and Conv2D + LSTM and then compare all the models based on training and validation accuracy.

The below table will provide the outcome of these experiments –

|  |  |  |  |
| --- | --- | --- | --- |
| EXPERIMENT | MODEL | RESULT | DECISION + EXPLANATION |
| 1 | Conv3D | Training Accuracy : 95%  Validation Accuracy : 39%  *(Best validation Accuracy,Epoch:9/25)* | Model is clearly overfit. Convolution layers with dropout layers and batch normalization are used. There were no augmentation and image enhancement done for images. We can increase batch size and use reduced dimension of image in our next experiment. |
| 2 | Training Accuracy : 95.7%  Validation Accuracy : 75% | Model is still overfit. This model has higher batch size of 40, 30 epochs with single channel and reduced dimension of image. This helped improve the validation accuracy to 75% compared to model in experiment #1 as only 1 channel data was used. |
| 3 | Training Accuracy : 68.2%  Validation Accuracy : 59%  *(Best Accuracy,Epoch:26/30)* | Model accuracy dropped. This model has higher number of convolution layers compared to model in experiment #2 and used all 30 images from 3 channels. Clearly, the augmentation and image enhancement is required to improve accuracy. |
| 4 | Training Accuracy : 93%  Validation Accuracy : 92% | This is an excellent model using Conv3D. Lower batch size and more layers with augmentation and image enhancement resulted in higher learning time per epoch. When batch size was increased beyond 10 then the model failed to train due to memory issues. |
| 5 | Conv2D + GRU | Training Accuracy : 90% Validation Accuracy : 64% | The model is overfit. This is Conv 2D and Gated Recurring Unit (GRU) based model with image of 120 x 120 with 3 channels. Let us try LSTM instead of GRU in our next experiment to improve the model’s validation accuracy. |
| Final Model | Conv2D + LSTM | Training Accuracy : 96.47% Validation Accuracy : 91%  *(Best validation Accuracy,Epoch:42/50)* | This is the best model for the Gesture Recognition considering the validation accuracy of 91%. Comparing it with the Conv3D model, it has far lesser training parameters(35,413) vs (3,100,869) and took less time to train. |