## **Deep Learning Course Project- Gesture Recognition**

**Teekam Khandelwal - Group Facilitator**

**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.

The gestures are continuously monitored by the webcam mounted on the TV. Each gesture corresponds to a specific command:

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

**Data Sets present in below link –**

<https://drive.google.com/uc?id=1ehyrYBQ5rbQQe6yL4XbLWe3FMvuVUGiL>

**Testing optimum value for batch size:** First we experimented with different batch sizes. The batch size of 128 causes the error to be thrown so we did our testing only with batch size of 32 and 64.

**For image resolution** we have tested the models with 80\*80 resolution and tried the model with 120\*120 resolution. We were not able to get much add to accuracy with higher resolution image, so we stuck with 80\*80 image for the final model.

**NOTE: -** The first model which threw generator error and the memory error, we built later top of that, so the notebook contains our third model as the first model which was ran successfully using Conv3D.

We are using sgd optimizer as have experienced it would well with the con2D models. We expect it to perform well with this conv3D model as well.

The following table consists of the experiments done to build a model to predict the gestures from the given data set:

|  |  |  |  |
| --- | --- | --- | --- |
| **Experiment Number** | **Model** | **Result** | **Decision + Explanation** |
| **1 & 2** | **Conv3D** | **Model 1 - threw out of memory error with batch size 128. So, we going further with 32 & 64 batch size.**  **Model 2 -Throws Generator error** | **Batch size – to be used 32 & 64(for some models).**  **Image resolution- to be used at 80\*80,& 120\*120 if it improves the accuracy with increasing batch size and resolution.** |
| **3-> First model in notebook (Built on top of 1 & 2)** | **Conv3D** | **No of Params =** 1,733,509  **Epochs = 20**  **Training Accuracy = 0.8021**  **Validation Accuracy = 0.3750** | **The Validation accuracy was very les respect to the train accuracy. Model is overfitting.**  **We decided to test if we could get best results if we change the batch size to 64. We will also increase the epochs to see if the validation accuracy to get how to affects the validation accuracy.** |
| **4 -> Model 2** | **Conv3D** | **No of Params =** 1,733,509  **Epochs = 30**  **Batch Size=64**  **Training Accuracy = 0.99**  **Validation Accuracy = 0.80** | **We find the validation accuracy is improved with the increased number of epochs & batch size.**  **Now we can experiment if we can get better results by improving the image resolution. Previously we tried with the 80\*80 image resolution and only 13 images from the set we considered. Now we will use 120\*120 image resolution and we will use 20 images from each video frames.** |
| **5 -> Model 3** | **Conv3D** | **No of Params =** 7,467,909  **Epochs = 20**  **Training Accuracy = 0.87**  **Validation Accuracy = 0.398** | **In this model we can see the model is overfitting as the validation accuracy is dropped drastically.**  **We will now experiment if increasing the no of epochs to 40 help to increase the validation accuracy as we saw in the first two models. .** |
| **6 -> Model 4** | **Conv3D** | **No of Params =** 7,467,909  **Epochs = 40**  **Training Accuracy = 0.91**  **Validation Accuracy = 0.82** | **Here we see the validation accuracy is increased compared to the previous model, but respect to the training accuracy it was less. This concludes no of epochs increased do not help much in improving the model which is overfitting.**  **Now we will try to implement the CNN + LSTM model and will compare the results with the second model which had highest validation accuracy but in model2 also validation accuracy have lees then training accuracy.** |
| **7 -> Model 5** | **CNN+LSTM** | **No of Params =** 6,714,117  **Epochs = 20**  **Dropout=0.25**  **Training Accuracy = 0.78**  **Validation Accuracy = 0.57** | **This model is having overall less training and validation accuracy, but we see steady trend in increasing the training and validation accuracy. Adding more epochs can help here to further improve the accuracy.** |
| **8 -> Model 6 -> Final Model** | **CNN+LSTM** | **No of Params =** 6,714,117  **Epochs = 40**  **Dropout = 0.2**  **Training Accuracy = 0.70**  **Validation Accuracy = 0.80** | **On further training this model did reach good training as well as validation accuracy.** |