**Neural Networks Project - Gesture Recognition**

**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 **recognise 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:

* 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

**Understanding the Dataset**

The training data consists of a few hundred videos categorised 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.

**Goals of this Project**

In this project, we have to build a model to recognise 5 hand gestures. The starter code has been shared with us

We need to accomplish the following in the project:

1. **Generator:**  The generator should be able to take a batch of videos as input without any error. Steps like cropping, resizing and normalization should be performed successfully.
2. **Model:** Develop a model that is able to train without any errors which will be judged on the total number of parameters (as the inference(prediction) time should be less) and the accuracy achieved.
3. **Write up:** This should contain the detailed procedure followed in choosing the final model. The write up should start with the reason for choosing the base model, then highlight the reasons and metrics taken into consideration to modify and experiment to arrive at the final model.

**Observation-**

* It was observed that as the Number of trainable parameters increase, the model takes much more time for training.
* **Batch size ∝ GPU memory / available compute.** A large batch size can throw *GPU Out of memory error,* and thus here we had to play around with the batch size till we were able to arrive at an optimal value of the batch size which our GPU could support
* We observed that batch size is not impacting training time but number of frames did. We will use batch size of 64 as it seems optimal
* We observed Con2d+LSTM gave the best validation accuracy in comparison to Con3d
* For detailed information on the Observations refer below table-

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MODEL** | **EXPERIMENT** | **RESULT** | **DECISION + EXPLANATION** | **PARAMETERS** |
| **Conv3D** | **1** | **OOM Error** | **Reduce the batch size and set frames at 16 with epochs 20** | **-** |
| **2** | **Training Accuracy : 0.91**  **Validation Accuracy : 0.50** | * **Overfitting** * **Let’s increase the frames to 30 and epochs to 30** | **2,067,621** |
| **3** | **Training Accuracy : 0.97**  **Validation Accuracy : 0.875** | * **Overfitting is reduced, but lot of parameters** * **Lets reduce the parameters by adding padding in con3D layers reducing filter size to (3,3,3)** | **9,006,245** |
| **4** | **Training Accuracy : 0.94**  **Validation Accuracy : 0.50** | * **Parameters has reduced but overfitting is introduced .** * ***Let's trying decreasing frames to 20*** | **5,618,245** |
| **5** | **Training Accuracy : 0.92**  **Validation Accuracy : 0.375** | * **Overfitting is further increased** * **Lets try adding BatchNormalization before MaxPooling and reduce kernel to (2,2,2)** | **1,907,909** |
| **6** | **Training Accuracy : 0.97**  **Validation Accuracy : 0.50** | * **Overfitting is decreased** * **Lets switch to Con2D+LSTM** | **1,301,045** |
| **CNN+LSTM** | **7**  **(Model-6 on Notebook)** | **Training Accuracy : 0.97**  **Validation Accuracy : 1.00** | **CNN - LSTM model - we get a best validation accuracy of 100%.** | **3,084,133** |
| **Transfer Learning using MobileNet** | **8 Training Accuracy : CNN - LSTM model with reduced parameters 2,866,533**  **0.96**  **Validation Accuracy :**  **0.87** | | | |
| **9** | **Training Accuracy : 0.98**  **Validation Accuracy : 0.56** | **CNN - LSTM model with frames reduced to 16** | **2,866,533** |
| **10** | **Training Accuracy : 0.995**  **Validation Accuracy : 0.75** | ***We are not training the MobileNet weights that we can see- validation accuracy is very poor.*** | **4,611,781** |