**Neural Networks-Gesture Recognition**

By Divya N L

**Problem Statement:**

A home electronics company wants to develop a cool feature to their smart TV that can recognize five different gestures performed by the user which helps user control the TV without remote. A model which can accurately detect the five gestures has to be developed.

The five gestures are as follows:

* 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

**Dataset:**

Dataset consists of total 773 videos, 663 for train and 100 for validation. These videos are categorized into one of the five classes. Each video is divided into sequence of 30 frames. 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. Specifically, videos have two types of dimensions - either 360x360 or 120x160.

**Model Building and Training:**

* Conv3D and CNN+RNN are the two types of architectures used for video classification tasks. 3D convolutions apply a 3-dimensional filter to the dataset and the filter moves in 3-directions (x, y, z) to calculate the low-level feature representations. CNN+RNN uses 2D convolutional layers followed by a GRU or LSTM layers.
* Initially a baseline model was built to check if the generator function is functional and model is running with no errors. Baseline model had two 3D convolutional layers with 32 feature maps ,64 feature maps and 256 feature maps and one dense layer(relu) followed by an output SoftMax layer.
* Then, a conv3d model was built with 32 feature maps ,64 feature maps,128 feature maps, 256 feature maps and 512 feature maps respectively and five dense layers(relu) followed by an output SoftMax layer.
* A **CONV2D + LSTM model was built** with 32 feature maps ,64 feature maps,128 feature maps, 256 feature maps and 512 feature maps respectively and five dense layers(relu) followed by an output SoftMax layer.
* Many experiments were conducted to get an optimal model in terms of good performance with less parameters and memory.
* In these experiments mainly image size, batch size, filter/kernel sizes and architectures were changed to get the optimal model.
* Batch size were fixed to 30experiments due to low availability of memory (out of memory).
* Also, number of epochs were fixed to 15due to huge computational time. Total of 18 frames per video were taken starting from 6th to 24th frame of each video

**Consolidated final models:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Model Name** | **No. of Total Params** | **No. of Trainable Params** | **No. of Non-Trainable Params** | **Augment Data** | **Drop out** | **Categorical Accuracy** | **Optimizers** | **Validation Categorical accuracy** |
| Conv3D | 52,202,981 | 52,202,597 | 384 | Yes | 0.5 | 0.1538 | ADAM | 0.1594 |
| **CONV2D + LSTM** | 6,817,829 | 6,817,829 | 0 | Yes | 0.5 | 0.1538 | SGD | 1.6829 |

