**Neural Networks Group Case Study: Hand Gesture Recognition Project**

***Presented by: Vishwanath Madhu& Vivek***

**Problem Statement**

As a data scientist at a home electronics company specializing in advanced smart televisions, the objective is to develop a feature enabling the recognition of five distinct gestures performed by users. This feature aims to facilitate TV control without the need for a remote. The gestures, which include thumbs up, thumbs down, left swipe, right swipe, and stop, correspond to specific commands such as volume adjustment, playback control, and skipping within a video.

The dataset comprises several hundred videos, each categorized into one of the five gesture classes. Each video consists of a sequence of 30 frames captured by a webcam mounted on the TV. The training data is organized into 'train' and 'val' folders, with corresponding CSV files containing information about the videos and their labels. Additionally, the data is stored in subfolders, each representing a video of a particular gesture.

The videos exhibit varying dimensions, either 360x360 or 120x160, depending on the webcam used for recording. Therefore, preprocessing is necessary to standardize the videos before model training. Each row in the CSV files represents a video and includes the subfolder name containing the video frames, the gesture label, and a numeric label (ranging from 0 to 4).

The task expects training a model on the 'train' folder to perform well on the 'val' folder, adhering to standard practices in machine learning projects. The test folder is withheld for evaluation purposes, and the final model's performance will be assessed on this set.

|  |  |  |  |
| --- | --- | --- | --- |
| **Experiment Number** | **Model** | **Result** | **Decision + Explanation** |
| **1** | **Conv3D** | **Trainable Params =** 2,477,669  **Epochs = 20**  **batch size = 20**  **Training Accuracy = 90.4%**  **Validation Accuracy = 73%** | Here we are using convolution 3D architecture with 2 conv3d layer with 16 and 32 kernels of size 3x3x3. Image size is 80x80. |
| **2** | **Conv3D** | **Trainable Params =** 9,897,669  **batch size = 20**  **Epochs = 25**  **Training Accuracy = 90.1%**  **Validation Accuracy = 79%** | Next, we tried with convolution 3D architecture with 2 conv3d layer with 32 and 64 kernels of size 3x3x3 to achieve better acuracy . Image size is 80x80.  We see there is an improvement in Validation accuracy. |
| **3** | **Conv3D** | **Trainable Params =** 5,549,669  **batch size = 20**  **Epochs = 20**  **Training Accuracy = 91.9%**  **Validation Accuracy = 77%** | Here we are using convolution 3D architecture with 2 conv3d layer with 16 and 32 kernels of size 3x3x3 .But here we are using Image size is 120x120 so as to check if we can improve performance of model 1 by increasing image resolution and making it lighter than model 3. We see that even though we have bit less validation accuracy than model 2 but its nearly half size than model 2 |
| **4** | **CNN 3D+LSTM** | **Trainable Params =**13,240,133  **batch size = 20**  **Epochs = 25**  **Training Accuracy = 84.7%**  **Validation Accuracy = 77%** | Here we are using convolution 3D +LSTM architecture with 2 conv3d layer with 32 and 64 kernels of size 3x3x3. And LSTM with 128 neurons. Using Image size is 80x80. We see that this model is more stable in terms of validation and training accuracy, but it is way complex and heavier than models we experimented till now. |
| **5** | **CNN 2D+LSTM** | **Trainable Params =** 1,811,077  **batch size = 20**  **Epochs = 25**  **Training Accuracy = 61%**  **Validation Accuracy = 59%** | Here we are using 2D CNN + LSTM architecture to process the video frames. It has 4 Convolution layers with 16,32,64 and 128 kernels of (3x3) and a LSTM layer with 128 neurons. Using Image size as 80x80 |
| **6** | **CNN 2D+LSTM** | **Trainable Params =** 6,722,053  **Batch size = 30**  **Epochs = 30**  **Training Accuracy = 66.7%**  **Validation Accuracy = 67%** | This is 2D CNN + LSTM architecture to process the video frames. It has 3 Convolution layers with 32,64 and 128 kernels of (3x3) and a LSTM layer with 128 neurons. Using Image size as 80x80.Increasing epoch to 30. Here we se that Mode; got heavier but still the model is under performing and is less efficient than model 1,2 3 and 4. |
| **Final Model** |  | **Based on above 6 experiments, Model 3 is better suited with its decent accuracy and number of trainable parameters.**  **Trainable Params =** 5,549,669  **batch size = 20**  **Epochs = 20**  **Training Accuracy = 91.9%**  **Validation Accuracy = 77%** |  |