Neural Networks Group Case Study: Gesture Recognition Project

Problem Statement:

This project involves building a 3D Convolutional Neural Network (CNN) to correctly recognize hand gestures by a user to control a smart TV.

The objective of this projects is to build a hand gesture recognition model that can be hosted on a camera installed in a smart TV that can understand 5 gestures.

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

About 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.

The videos have two types of dimensions - either 360x360 or 120x160 (depending on the webcam used to record the videos).

Data Source: https://drive.google.com/uc?id=1ehyrYBQ5rbQQe6yL4XbLWe3FMvuVUGiL

Neural Network Architectures Used:

For analysing videos using neural networks, we have used below models :

1) Conv3D
2) Time Distributed Conv2D + GRU
3) Time Distributed Conv2D + GRU (Dropout=0.2)
4) Time Distributed Conv2D + Dense
5) Time Distributed + ConvLSTM2D

Experiment Number	Model	Result	Decision + Explanation
1	Conv3D	Train Accuracy: 0.6369, Validation Accuracy: 0.6562	The model is good and the training and validation scores are good. The model has 710,533 trainable parameters. Its performance will increase if increase number of epochs.
2	Time Distributed Conv2D + GRU	Train Accuracy: 0.8482, Validation Accuracy: 0.4844	The model is working quite well on training dataset with less trainable parameters (98,885). Let's add some drop outs after each layer, so that both train and validation accuracies will be closure.
3	Time Distributed Conv2D + GRU (Dropout=0.2)	Train Accuracy: 0.8169, Validation Accuracy: 0.4531	The model accuracy further deteriorated; Let us replace GRU with a plain Dense Layer Network and some Global Avg Pooling.
4	Time Distributed Conv2D + Dense	Train Accuracy: 0.8646, Validation Accuracy: 0.5625	This is good model with training and validation accuracies with number of params 128,517. Let us use different architecture of model with time distributed and ConvLSTM2D. Its performance will increase if increase number of epochs.
5	Time Distributed + ConvLSTM2D	Train Accuracy: 0.6845, Validation Accuracy: 0.6406	This is the best model so far we can get. The validation accuracy is good and the numbers of parameters are 13,589. The model size is also so small 226KB. Its performance will increase if increase number of epochs.

Conclusion:

The Model built with Time distributed Conv2D and ConvLSTM2D (Experiment #5) gave similar results compared to Conv3D (Experiment #1).

We are choosing Time distributed Conv2D and ConvLSTM2D (Experiment #5) as best models because it has less numbers of parameters, 13,589. Also, the model size is small 226KB.

Whereas Conv3D (Experiment #1) has 710,533 trainable parameters with 8.5 MB model size.

Performance of all models will increase with increase in number of epochs.