Gesture Recognition Using Neural Networks – Project Write-up

Created by: Anupam Maiti

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 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. Each video is a sequence of 30 frames (or images).

- 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 which are categorized into 5 classes. Each video is divided into 30 frames of sequence(images). The videos are recorded by different people performing one of the 5 gestures in front of a webcam. It's similar to what Smart TV will be using. We have provided with a .zip file contains train and test folders with two csv for the given folders.

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

Model Findings:

Exp No	Model	No of parameters	Hyperparameters	Result	Decision + Explanation
1	Conv3D	5,04,709	Number of sample frames = 16 Batch size = 10 Number of epochs = 20	Training accuracy: 0.87 Validation accuracy: 0.78 Indication of overfitting	We used a Conv3D setup to train on 16 frames from the video. Our training accuracy hit 87%, but validation only reached 78%, suggesting overfitting might be happening. To address this concern, we aim to experiment with an alternative architecture, combining CNN with LSTM.
2	CNN with LSTM	1,657,445	Number of sample frames = 18 Batch size = 20 Number of epochs = 20	Training accuracy: 0.98 Validation accuracy: 0.91 Indication of overfitting	The training accuracy keeps getting better, hitting 98%, but validation only peaks at 91%, still hinting at overfitting. To tackle this, we're thinking of trying out a different setup, mixing CNN with GRU, while

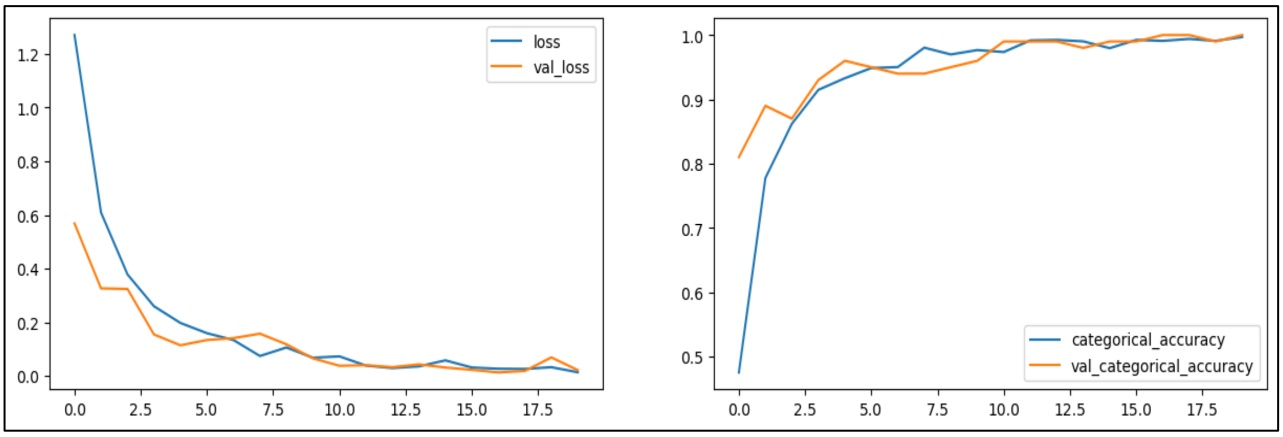
					sticking to the same number of frames.
3	CNN with GRU	2,573,925	Number of sample frames = 18 Batch size = 20 Number of epochs = 20	Training accuracy: 0.99 Validation accuracy: 0.88 Indication of overfitting	By adding GRU-RNN, the training accuracy edged up to 99%, but validation dipped to 88%, still showing signs of overfitting. Next, we aim to explore whether we can enhance the training and validation accuracy by reducing the parameter count using GRU and leveraging the trainable weights of MobileNet.
4	CNN with GRU (with trainable weights of Transfer Learning)	3,228,864	Number of sample frames = 16 Batch size = 5 Number of epochs = 20	Training accuracy: 0.99 Validation accuracy: 0.99 Finally, model based on accuracy metric	Using GRU and transfer learning with pre-trained weights, we see a significant jump in training accuracy to 99% and validation accuracy to 99%. This shows we've effectively tackled overfitting, and the model performs strongly on the validation set. So, we can consider this our final model for evaluation.

Conclusion:

Based on the results CNN with GRU (with trainable weights of Transfer Learning) model performing well on the dataset provided for Gesture Recognition.

Selected model: **CNN with GRU (with trainable weights of Transfer Learning)**

- Training Accuracy: 0.99
- Validation Accuracy: 0.99
- Batch Size: 5
- Frames: 16
- Number of epochs: 20
- Model File Name: model-00020-0.01408-0.99698-0.02186-1.00000.keras
- Model Loss and Accuracy comparison b/w Train and Validation set:


loss
val_loss
categorical_accuracy
val_categorical_accuracy