**Gesture Recognition**

- By Prachi Shah and Pooja Kamath

* **Overview**

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 gesturesperformed 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' backwards10 seconds
* Right swipe: ’Jump' forward 10 seconds
* Stop: Pause the movie

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Exp No** | **Model** | **Number of parameters** | **Hyperparameters** | **Result** | **Decision + Explanation** |
| 1 | CNN with LSTM | 1,657,445 | Number of sample frames = 30  Batch size = 20  Number of epochs = 20 | Training accuracy: 0.88  Validation accuracy: 0.73  Indication of overfitting | Using the base CNN with LSTM architecture, we trained on all 30 frames extracted from the video. The model achieved a training accuracy of 88% and a validation accuracy of 73%, suggesting potential overfitting. To mitigate this, we plan to modify the model by reducing the number of frames, batch size, and epochs. |
| 2 | CNN with LSTM (with reduced hyperparameters) | 1,657,445 | Number of sample frames = 20  Batch size = 10  Number of epochs = 10 | Training accuracy: 0.81  Validation accuracy: 0.68  Indication of underfitting | Reducing the number of hyperparameters led to a drop in training accuracy to 81% and validation accuracy to 68%, indicating underfitting. To mitigate this, we will focus on reducing the number of frames while maintaining a batch size of 20 and training for 20 epochs. |
| 3 | CNN with LSTM (with reduced frames) | 1,657,445 | Number of sample frames = 18  Batch size = 20  Number of epochs = 20 | Training accuracy: 0.91  Validation accuracy: 0.75  Indication of overfitting | The training accuracy continues to improve, reaching 91%, while the validation accuracy peaks at 75%, still suggesting potential overfitting. To combat this, we plan to explore an alternative architecture by integrating CNN with GRU, keeping the same number of frames, to enhance generalization and performance. |
| 4 | CNN with GRU | 2,573,925 | Number of sample frames = 18  Batch size = 20  Number of epochs = 20 | Training accuracy: 0.87  Validation accuracy: 0.82  Indication of overfitting | Integrating GRU-RNN led to a slight drop in training accuracy to 87%, while validation accuracy declined to 82%, signaling overfitting. To address this, we plan to introduce additional augmentation techniques, such as rotation, and reattempt the same architecture to improve generalization and mitigate overfitting. |
| 5 | CNN with LSTM (with modified Augmentation Technique) | 1,657,445 | Number of sample frames = 18  Batch size = 20  Number of epochs = 20 | Training accuracy: 0.91  Validation accuracy: 0.75  Slightly improved with augmentation | After incorporating rotation in the augmentation phase, the training accuracy slightly dropped to 91%, while the validation accuracy improved to 77%, suggesting better generalization. Next, we will explore transfer learning to determine if it can further boost validation accuracy and enhance model performance. |
| 6 | CNN with LSTM (with Transfer Learning) | 3,840,453 | Number of sample frames = 16  Batch size = 5  Number of epochs = 20 | Training accuracy: 0.98  Validation accuracy: 0.77  Indication of overfitting | Implementing LSTM with transfer learning boosted the \*\*training accuracy to 98%\*\*, yet the \*\*validation accuracy remained at 77%\*\*, indicating potential overfitting. This may stem from the \*\*MobileNet weights not being trained\*\*, limiting the model’s ability to generalize. Next, we will explore whether \*\*training MobileNet’s weights alongside LSTM\*\* can address this issue and improve overall performance. |
| 7 | CNN with LSTM (with trainable weights of Transfer Learning) | 3,840,453 | Number of sample frames = 16  Batch size = 5  Number of epochs = 20 | Training accuracy: 0.98  Validation accuracy: 0.95  Improved accuracy but model training was computationally expensive | After training the MobileNet weights, the training accuracy surged to 98%, with validation accuracy reaching 95%. However, this gain came at the expense of significantly longer training time due to a fourfold increase in parameters within the LSTM architecture. Moving forward, we will investigate whether reducing the parameter count with GRU while still leveraging MobileNet’s trainable weights can further enhance accuracy and efficiency. |
| Final Model | CNN with GRU (with trainable weights of Transfer Learning) | 3,693,253 | Number of sample frames = 16  Batch size = 5  Number of epochs = 20 | Training accuracy: 0.9940  Validation accuracy: 0.97  Finally, model based on accuracy metric | By leveraging GRU and transfer learning with pre-trained weights, our model achieves an impressive 99% training accuracy and 98% validation accuracy. This marks a significant breakthrough, effectively addressing overfitting while ensuring robust generalization. Given its strong performance, we confidently finalize this model for evaluation. |