**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 recognise 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:

* 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

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

**About dataset:**

The training data consists of a few hundred videos categorized 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 like what the smart TV will use.

**List of Models and findings:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Experiment Number** | **Model** | **Result** | **Decision + Explanation** |
| **1** | **Conv2D layers + GRU Layer + No dropout** | * Training Accuracy: 0.96 * Validation Accuracy: 0.59 | * Clear indication of overfitting since there is huge difference between training accuracy and validation accuracy * We will ignore this model for now and try next model to handle overfitting by using batch normalization and dropouts |
| **2** | Conv2D layers + GRU Layer + dropout | * Training Accuracy: 0.81 * Validation Accuracy: 0.63 | * Dropout was added after every convolution unit and GRU unit. * The overfitting has slightly reduced compared to the previous model * As this model is still overfitted, so we will use L2 regularization to remove overfitting |
| **3** | Conv2D layers + GRU Layer + dropout + L2 regularization | * Training Accuracy: 0.78 * Validation Accuracy: 0.51 | * There is slight improvement in the overfitting after introducing L2 regularization and dropout * We will use LSTM layer in the next model to see if accuracy is improved and if helps in overfitting. |
| **4** | Conv2D layers + LSTM Layer + dropout + L2 regularization | * Training Accuracy: 0.77 * Validation Accuracy: 0.51 | * There seems to be no improvement in the model. * We will experiment with Transfer learning in the subsequent models |
| **5** | Transfer Learning + RestNet50 + GRU Layer | * Training Accuracy: 0.26 * Validation Accuracy: 0.22 | * This model has both poor training and validation accuracy. * The model is overtly expensive both in terms of computation and memory. * We will experiment with mobilenet in next run. |
| **6** | Transfer Learning + mobilenet + GRU Layer | * Training Accuracy: 0.93 * Validation Accuracy: 0.83 | * The training accuracy and the validation accuracy have increased substantially after using mobilenet * We will increase the number of epoch to further stabilize and improve the scores. * Use Mobilenet as it is lightweight in its architecture. |
| **7** | Transfer Learning + mobilenet + GRU Layer + Increased epoch | * Training Accuracy: 0.93 * Validation Accuracy: 0.84 | * A slight improvement in the validation accuracy can be seen in this model * We will increase epoch and batch size in the next model for further improvement |
| **8** | Transfer Learning + mobilenet + GRU Layer + Increased epoch + Increase Batch Size | * Training Accuracy: 0.97 * Validation Accuracy: 0.95 | * So far, this is the best model we got in terms of training and validation accuracy * Increase both epoch and batch size results in higher learning rate and test accuracy. * This model is stable, and we will consider as our final model. * Transfer learning is flexible, allowing the use of pre-trained models directly, as feature extraction preprocessing, and integrated into entirely new models. |
| **9** | Conv3D layers + BatchNormalization | * Training Accuracy: 0.65 * Validation Accuracy: 0.58 | * So far we have tried CNN + RNN architecture. We will try few Con3D models. * Although this model is not overfitted, but the accuracy score is poor * We will use dropout and L2 regularization on Conv3D in the subsequent models to improve the accuracy scores |
| **10** | Conv3D layers + Dropout + BatchNormalization + L2 Regularization | * Training Accuracy: 0.65 * Validation Accuracy: 0.61 | * As we can see, both training and validation accuracy have improved compare to the previous model * We will experiment with adding more layers in the next model |
| **11** | Conv3D layers + More Layers + Dropout + L2 Regularization | * Training Accuracy: 0.21 * Validation Accuracy: 0.29 | * Addition of layers resulted in decrease in the accuracy of the model * Let us add more layers and observe the difference |
| **12** | Conv3D layers + More Layers + Dropout + L2 Regularization | * Training Accuracy: 0.35 * Validation Accuracy: 0.33 | * The validation accuracy improved after adding more layers * We will go for larger dense layers in the next model to improve the score |
| **13** | Conv3D layers + Larger Dense Layer + Dropout + L2 Regularization | * Training Accuracy: 0.36 * Validation Accuracy: 0.44 | * In this model, the overfitting has been minimized than that of the previous model * We will build the next model without L2 regularization and observe the change |
| **14** | Conv3D layers + Larger Dense Layer + Dropout | * Training Accuracy: 0.64 * Validation Accuracy: 0.29 | * Reducing the number of network parameters by removing L2 regularization resulted in improvement in training accuracy and substantial reduce in validation accuracy i.e. the model is overfitted |

**Conclusion:**

This project involves multiple deep learning models to understand and accurately predict the gestures performed by the users. Below are some key points which we noticed and that helped us conclude the project.

* At first from the experiments we noticed that usage of L2 regularization helped us reduce overfitting and model seemed to be pretty stable. However, we had to increase the accuracy.
* After that, we tried using transfer learning along with GRU layer and on increasing batch size and number of epoch, we found that the accuracy was increased with a surge in learning rate as in Model no. 8, this is the overall best model with highest training and validation accuracy.
* Then on trying out with Conv 3D layers the model’s results seemed to be pretty much stable although the accuracy was not impressive, but after introducing L2 regularization with batch normalization we found that the accuracy slightly increased and was the best stable model in Conv 3D category for this project.