**Gesture Recognition – Deep learning**

**Problem Statement:** We need 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 following table consists of the experiments done to build a model to predict the gestures from the given data set.

|  |  |  |  |
| --- | --- | --- | --- |
| **Experiment Number** | **Model** | **Result** | **Decision + Explanation** |
| **1** | **Conv3D:** Simple Conv3D with 2 layers 2 Dense layers  Batch size =32  Optimiser = adam  Number of epochs=20 | **Train accuracy:69.20% Validation accuracy:55.47%** | **Model clearly shows Overfitting.** The difference between training and validation accuracy tends to increase as the epochs progress (e.g., from Epoch 10 onwards). This suggests that the model may be overfitting to the training data, as it performs significantly better on the training set compared to the validation set. |
| **2** | **Conv3D:** 3 Layers  2 Dense Layers  2 Dropouts at Dense layers  Batch size =32  Optimiser = adam  Number of epochs=20 | **Train accuracy:83.48% Validation accuracy:72.66%** | **Overfitting:** The difference between training and validation accuracy is relatively small in the early epochs but grows larger in later epochs (e.g., Epoch 20 has a difference of 0.1082).  This suggests that the model may be starting to overfit to the training data, as it performs significantly better on the training set compared to the validation set in later epochs.  Validation Accuracy Fluctuations:  Validation accuracy fluctuates slightly acrossepochs (e.g., drops at Epoch 17 and Epoch 20), which could indicate instability in the model's generalization performance. |
| **3** | **Conv3D: :** 3 Layers  2 Dense Layers  2 Dropouts at Dense layers  Batch size =64  Optimiser = SGD  Number of epochs=20 | **Train accuracy: 56.03% Validation accuracy:56.25%** | **Increased batch size =64 and number of epochs =30 Model result shows**: The model shows gradual improvement in both training and validation accuracy, but the performance is inconsistent, and convergence is slow. |
| **Best Model – Conv3D architecture** | **Conv3D:** 3 Layers  2 Dense Layers  2 Dropouts at Dense layers  Batch size =32  Optimiser = adam  Number of epochs=40 | **Train accuracy: 88.39% Validation accuracy:79.69%** | **Tried with number of epochs =40 with batch size=32**  **Model Result:** The model shows consistent improvement in both training and validation accuracy, but signs of overfitting emerge in later epochs **so far this Conv3D model gave best accuracy.** |
|  |  |  |  |
| **5** | **Conv2D LSTM:** | **Train accuracy: 98.21% Validation accuracy:71.88%** | Started with Conv2D + LSTM Model with 3 conv 2D layers , 2 LSTM Layers with 2 dense  Model accuracy summary: The large gap between training and validation accuracy (e.g., 98.21% vs. 71.88%) suggests that the model is overfitting to the training data.  The validation accuracy plateaus around 70-72%, indicating that the model struggles to generalize to unseen data.   The learning rate adjustments helped stabilize training but did not significantly improve validation performance. |
| **6** | **Conv2D GRU** | **Train accuracy: 100% Validation accuracy:64.84%** | Tried with Conv2D +GRU approach: Model result: Overfitting The training accuracy reached 100%, while the validation accuracy plateaued around 60-65%. |
| **7** | **Conv2D LSTM** | **Train accuracy: 88.84% Validation accuracy:60.16%** | **Tried with the less filters at each Conv2D layers but still the model result is showing Overfitting** Let’s try the same with Model with GRU to see if any increase in Validation accuracy |
| **Best Model – CNN + RNN architecture** | **Conv2D GRU** | **Train accuracy: 98.96% Validation accuracy:73.44%** | **Tried with the less filters at each Conv2D layers + GRU but still the model result is showing Overfitting** as compared to the above models, this model Validation accuracy slightly increased but still not generalizing well on un-seen data. Suggestion: The validation accuracy may increase if applies  1. Data Augmentation  2. Batch Normalization |
| **9** | **Conv3D:Created new generator function with images to process =20** | **Train accuracy: 83.48% Validation accuracy:35.94%** | **Overfitting:**  The model performs well on the training data (83.48% accuracy) but struggles on the validation data (35.94% accuracy).  This suggests that the model is memorizing the training data rather than generalizing to unseen data**.** |
| **10** | **Image Data Augmentation-Conv3D** | **Train accuracy: 61.84% Validation accuracy:64.00%** | Applied Image Data augmentation on Data Overall Model performance: The model steadily learned, with both training and validation accuracy improving over time but still had room for improvement. |