**Comparative Analysis of CNN and RNN Architectures for Gesture Recognition:**

| **Experiment Number** | **Model** | **Result** | **Decision + Explanation** |
| --- | --- | --- | --- |
| **1** | **Conv3D** | **Training Accuracy: 0.9969834089279175**  **Validation Accuracy: 0.4399999976158142** | **This model had a good training accuracy but a poor validation accuracy, indicating overfitting. The model was not able to generalize well to unseen data. This could be due to the smaller number of frames (16) and potentially insufficient training epochs. To improve, consider increasing the number of frames or epochs or experimenting with regularization techniques.** |
| **2** | **Conv3D** | **Training Accuracy: 0.9984917044639587, Validation Accuracy: 0.7799999713897705** | **With an increased number of frames (30) and epochs (30), the model showed improvement in validation accuracy. This indicates better generalization compared to Experiment 1. The higher validation accuracy suggests that the model is starting to capture the underlying patterns in the data better. This could be further optimized by fine-tuning the hyperparameters or experimenting with different pooling layers.** |
| **3** | **Conv3D with Reduced Parameters** | **Training Accuracy: 0.9894419312477112, Validation Accuracy: 0.36000001430511475** | **Reducing parameters and increasing image size to 160x160 did not yield better results. The validation accuracy dropped significantly, suggesting that the model became less capable of generalizing. This may be due to overfitting to the larger image size without sufficient regularization. It might be worth revisiting the number of parameters or trying different architectures.** |
| **4** | **Conv3D with Same Pooling Layers** | **Training Accuracy: 0.9683257937431335, Validation Accuracy: 0.6800000071525574** | **This experiment used 20 frames and the same pooling layers as before, but with a smaller batch size. The results show a decent validation accuracy, but it’s still not optimal. The decrease in training accuracy suggests that the model may not be learning enough features. Adjusting the number of layers or filters might help improve this model.** |
| **5** | **Conv3D with BatchNormalization Before MaxPooling** | **Training Accuracy: 0.9366515874862671, Validation Accuracy: 0.3799999952316284** | **Switching the BatchNormalization layer before MaxPooling did not improve the model. The training and validation accuracy both decreased, indicating that this architectural change negatively impacted the model's performance. This suggests that BatchNormalization should come after pooling layers or that other architectural changes might be needed to better capture the data's features.** |
| **6** | **Conv2D + LSTM** | **Training Accuracy: 1.0, Validation Accuracy: 0.6700000166893005** | **Switching to a Conv2D + LSTM architecture resulted in perfect training accuracy but a lower validation accuracy compared to Experiment 2. This indicates a strong overfitting issue, where the model memorizes the training data but fails to generalize. Reducing overfitting might require techniques like dropout, early stopping, or data augmentation.** |
| **7** | **Transfer Learning (MobileNet) with LSTM** | **Training Accuracy: 1.0, Validation Accuracy: 0.7599999904632568** | **Applying transfer learning with MobileNet and LSTM yielded slightly better validation accuracy than Experiment 6 but still fell short of Experiment 2. The use of MobileNet helps in leveraging pre-trained features, but the model might need more fine-tuning, such as adjusting the number of trainable layers or trying different LSTM configurations.** |
| **8** | **Conv2D + GRU** | **Training Accuracy: 1.0, Validation Accuracy: 0.6600000262260437** | **The Conv2D + GRU model performed similarly to the Conv2D + LSTM model in terms of validation accuracy. This suggests that while both GRU and LSTM architectures can fit the training data well, they may struggle with generalization when used with the Conv2D backbone. Further experimentation with regularization or different architectures could be beneficial.** |
| **9** | **Transfer Learning (MobileNet) with GRU** | **Training Accuracy: 1.0, Validation Accuracy: 0.9800000190734863** | **This model demonstrated the best generalization capability, with a validation accuracy of 0.9800. The combination of transfer learning with MobileNet and GRU outperformed all previous models, indicating that this approach effectively captures the temporal features of the data while leveraging the powerful pre-trained features from MobileNet. This model is recommended as the best-performing model, showing excellent balance between training and validation performance.** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| **Final Model** | **Transfer Learning (MobileNet) with GRU.** | **Training Accuracy: 1.0, Validation Accuracy: 0.9800000190734863** | **After evaluating multiple architectures and configurations, the final decision is to proceed with the Transfer Learning (MobileNet) with GRU model. This model showed the highest validation accuracy, indicating superior generalization to unseen data. It leverages the strengths of both pre-trained features and GRU's capability to model sequential data, making it the best choice for the task at hand.** |