Detailed experimental analysis table:

|  |  |  |  |
| --- | --- | --- | --- |
| **Experiment #** | **Model Architecture** | **Results** | **Decision + Explanation** |
| 1 | CNN-RNN (MobileNetV2 + GRU) | Initial val\_accuracy: 0.47  Final val\_accuracy: 0.86 | - Used MobileNetV2 as base CNN for feature extraction  - Added GRU layers for temporal modelling  - Model showed good learning progression  - Early stopping kicked in after 33 epochs |
| 2 | 3D CNN | Initial val\_accuracy: 0.21  Final val\_accuracy: 0.31 | - Pure 3D convolutions struggled with temporal patterns  - Model showed signs of underfitting  - Learning plateaued quickly  - Performance significantly worse than CNN-RNN |
| 3 | CNN-RNN (Fine-tuned) | Test accuracy: 0.82  F1-score: 0.81-0.93 across classes | - Fine-tuned with lower learning rate (1e-5)  - Unfroze later layers of MobileNetV2  - Achieved best performance on test set  - Strong per-class metrics |
| 4 | Data Augmentation | Training accuracy: 0.98  Validation accuracy: 0.86 | - Added frame sampling strategy  - Implemented proper image preprocessing  - Helped reduce overfitting  - Improved model generalization |

Final Model Selection & Rationale:

* Selected the fine-tuned CNN-RNN model combining MobileNetV2 and GRU layers
* Key advantages:
  1. Better temporal modeling through GRU layers vs pure 3D convolutions
  2. Transfer learning benefits from pretrained MobileNetV2
  3. Higher accuracy and F1-scores across all gesture classes
  4. More stable training progression

Architecture Details:

* Input shape: (30, 120, 120, 3) for 30 frame sequences
* MobileNetV2 base for spatial feature extraction
* Global average pooling to reduce dimensionality
* Dual GRU layers (256 & 128 units) with dropout
* Dense layers for final classification

Key Metrics:

* Training accuracy: 98%
* Validation accuracy: 86%
* Test accuracy: 82%
* Per-class F1-scores ranging from 0.81-0.93

The CNN-RNN architecture significantly outperformed the 3D CNN approach, likely due to:

1. Better handling of temporal dependencies through GRU layers
2. More efficient feature extraction via transfer learning
3. Better regularization through dropout and data augmentation
4. More stable gradient flow during training

Future Improvements:

1. Experiment with different backbone CNNs
2. Try attention mechanisms
3. Implement more aggressive data augmentation
4. Test different temporal modelling architectures