Human Activity Recognition Using ConvLSTM & LRCN

UCF50 Dataset | Akash Rana | April 2025

Agenda

- PROJECT MOTIVATION & PROBLEM STATEMENT
- DATASET & PRE-PROCESSING
- METHODOLOGY OVERVIEW
- MODEL ARCHITECTURES: CONVLSTM VS LRCN
- TRAINING CONFIGURATION
- EVALUATION & RESULTS
- COMPARATIVE INSIGHTS
- CONCLUSION & FUTURE WORK

Why Human Activity Recognition (HAR)?

KEY TO SURVEILLANCE, HEALTHCARE MONITORING, AND HCI.

NEED MODELS THAT CAPTURE BOTH SPATIAL AND TEMPORAL CUES IN VIDEO.

GOAL: CLASSIFY COMPLEX ACTIVITIES IN UNTRIMMED CLIPS EFFICIENTLY.

Broader Impact: Predictive Movement Modeling

- HAR outputs serve as inputs for HMM/DBN-based movement prediction.
- Towards proactive smart home automation: predicting future user actions.
- Synergy between deep learning (HAR) and probabilistic models (HMM, DBN).
- Enhances user-centric experiences through intelligent automation.

Dataset: UCF50

6,600+ YouTube clips across 50 activity classes.

Selected 4 diverse classes: WalkingWithDog, TaiChi, Swing, HorseRace.

Pre-processing: frame extraction (20 frames/clip), resize 64×64, normalization, one-hot labels.

Methodology Overview

Data preparation → Model design → Training → Evaluation.

Two deep architectures implemented:

- ConvLSTM – convolution inside recurrent gates.

- LRCN – CNN feature extractor + LSTM sequence model.

ConvLSTM Pipeline

Input tensor: (batch, 20, 64, 64, 3).

↓

Stack of ConvLSTM2D layers with 3×3 kernels.

↓

Temporal & spatial pooling via MaxPooling3D.

↓

Dropout for regularization.

↓

Flatten → Dense softmax (4 classes).

Layer (type)	Output Shape	Param #
conv_lstm2d (ConvLSTM2D)	(None, 20, 62, 62, 4)	1,024
max_pooling3d (MaxPooling3D)	(None, 20, 31, 31, 4)	0
time_distributed (TimeDistributed)	(None, 20, 31, 31, 4)	0
conv_lstm2d_1 (ConvLSTM2D)	(None, 20, 29, 29, 8)	3,488
max_pooling3d_1 (MaxPooling3D)	(None, 20, 15, 15, 8)	0
time_distributed_1 (TimeDistributed)	(None, 20, 15, 15, 8)	0
conv_lstm2d_2 (ConvLSTM2D)	(None, 20, 13, 13, 14)	11,144
max_pooling3d_2 (MaxPooling3D)	(None, 20, 7, 7, 14)	0
time_distributed_2 (TimeDistributed)	(None, 20, 7, 7, 14)	0
conv_lstm2d_3 (ConvLSTM2D)	(None, 20, 5, 5, 16)	17,344
max_pooling3d_3 (MaxPooling3D)	(None, 20, 3, 3, 16)	0
flatten (Flatten)	(None, 2880)	0

ConvLSTM model architecture

Total params: 44,524

Trainable params: 44,524

Non-trainable params: 0

LRCN Pipeline

1. TimeDistributed Conv2D (frame-level features).
2. MaxPooling + Flatten per frame.
3. LSTM (64 units) models temporal evolution.
4. Dense softmax classifier.
5. Modular: can swap in pretrained CNN backbones.

Layer (type)	Output Shape	Param #
time_distributed_3 (TimeDistributed)	(None, 20, 64, 64, 16)	448
time_distributed_4 (TimeDistributed)	(None, 20, 16, 16, 16)	0
time_distributed_5 (TimeDistributed)	(None, 20, 16, 16, 16)	0
time_distributed_6 (TimeDistributed)	(None, 20, 16, 16, 32)	4,640
time_distributed_7 (TimeDistributed)	(None, 20, 4, 4, 32)	0
time_distributed_8 (TimeDistributed)	(None, 20, 4, 4, 32)	0
time_distributed_9 (TimeDistributed)	(None, 20, 4, 4, 64)	18,496
time_distributed_10 (TimeDistributed)	(None, 20, 2, 2, 64)	0
time_distributed_11 (TimeDistributed)	(None, 20, 2, 2, 64)	0
time_distributed_12 (TimeDistributed)	(None, 20, 2, 2, 64)	36,928
time_distributed_13 (TimeDistributed)	(None, 20, 1, 1, 64)	0
time_distributed_14 (TimeDistributed)	(None, 20, 64)	0
lstm (LSTM)	(None, 32)	12,416
dense_1 (Dense)	(None, 4)	132

LRCN model architecture

Total params: 73,060

Trainable params: 73,060

Non-trainable params: 0

Training Setup

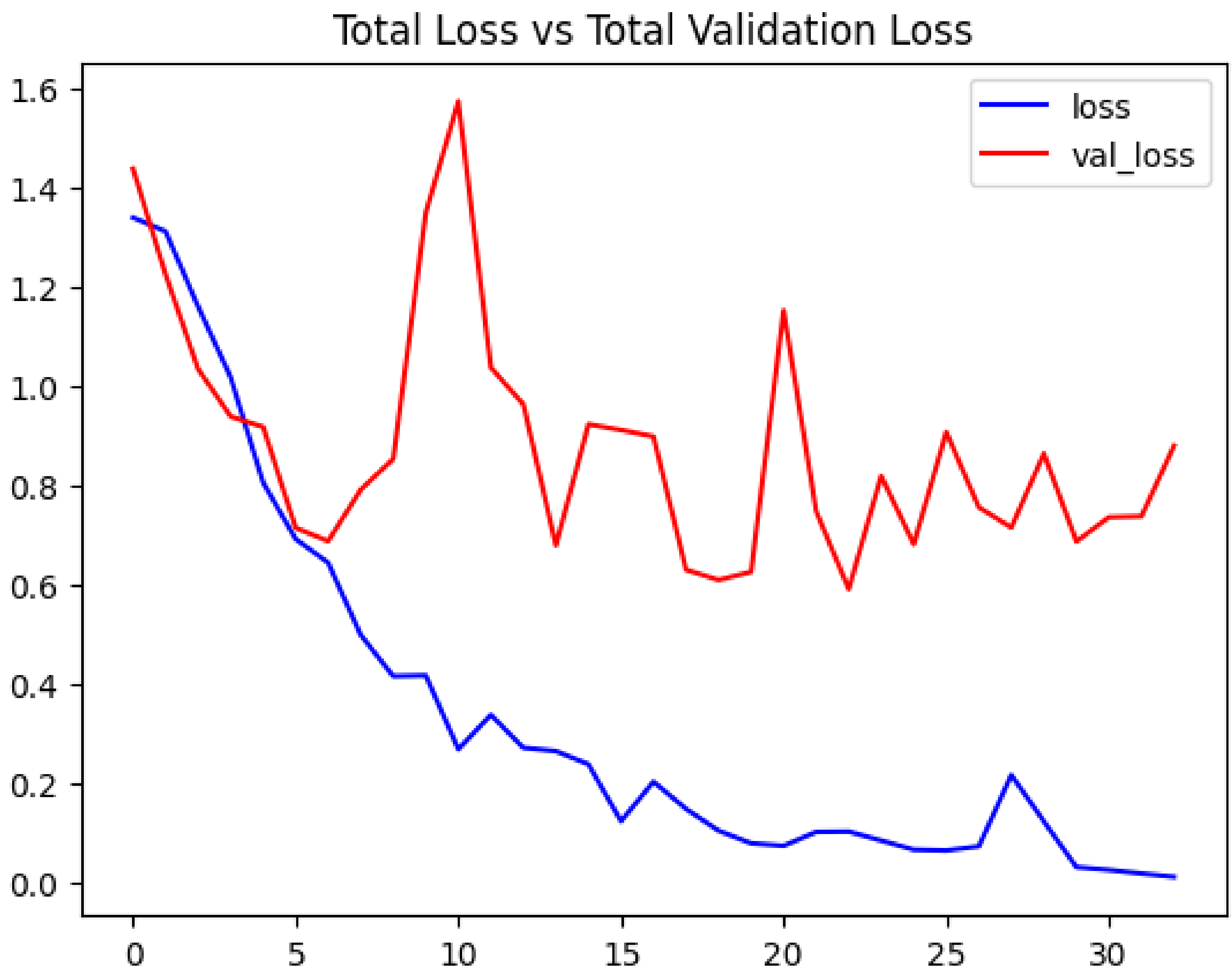
Loss: categorical cross-entropy | Optimizer: Adam

Batch size: 4 | Epochs: ≤ 30 (early stopping, patience = 5)

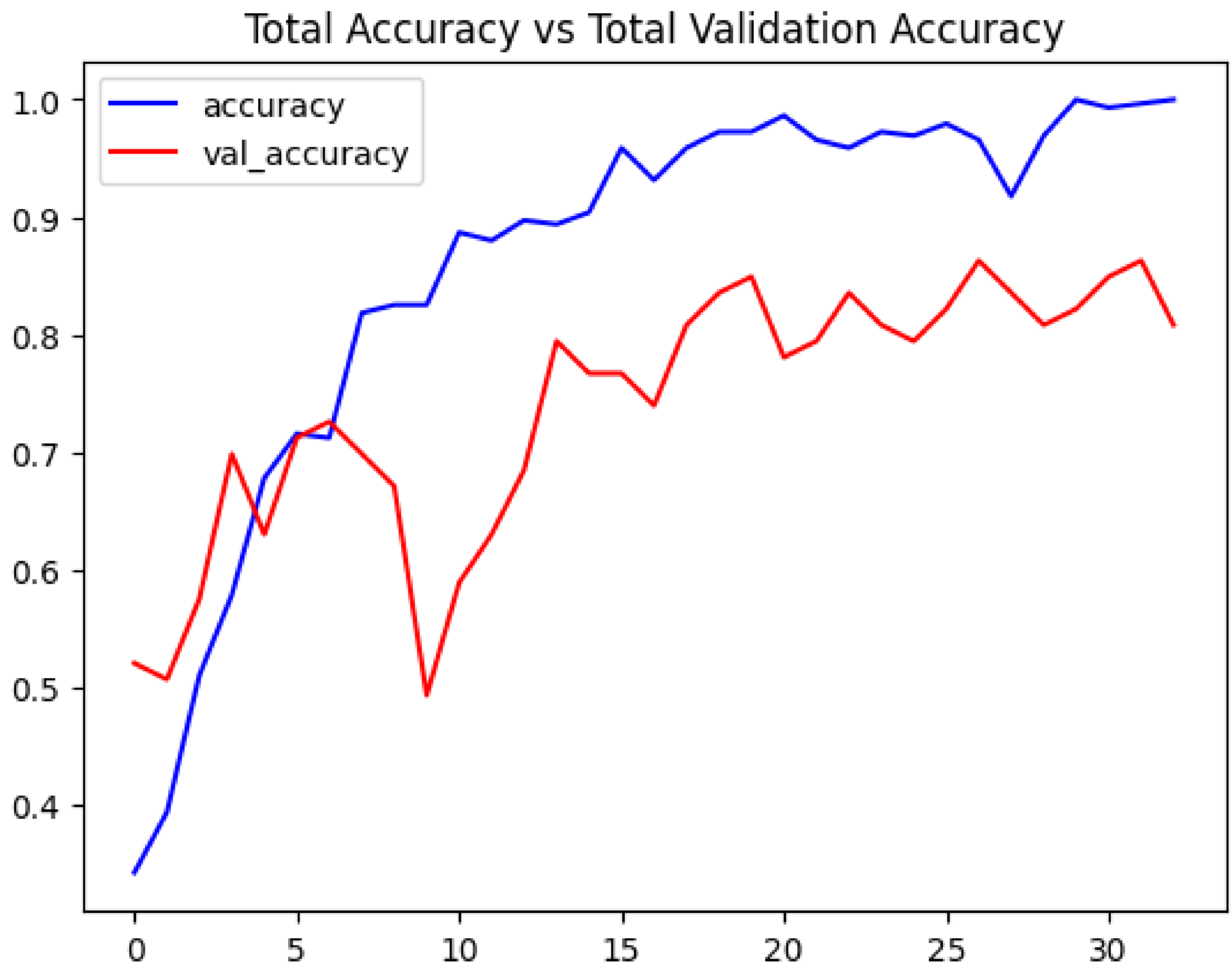
Validation split: 20% | Fixed random seed: 27

GPU-accelerated TensorFlow/Keras environment

ConvLSTM Results



ConvLSTM Results



ConvLSTM – Results

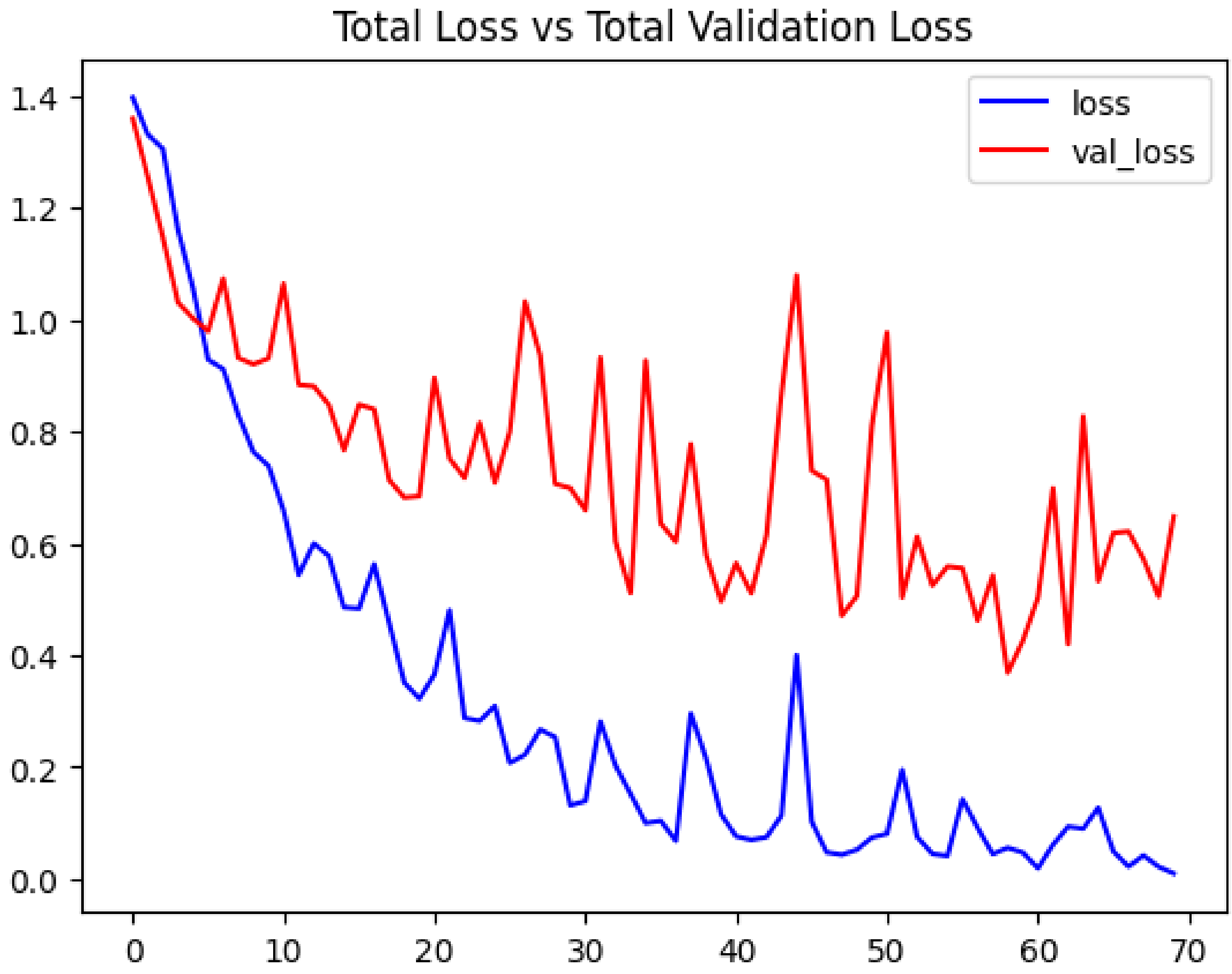
Test Accuracy: 79.25 %

Test Loss: 0.5773

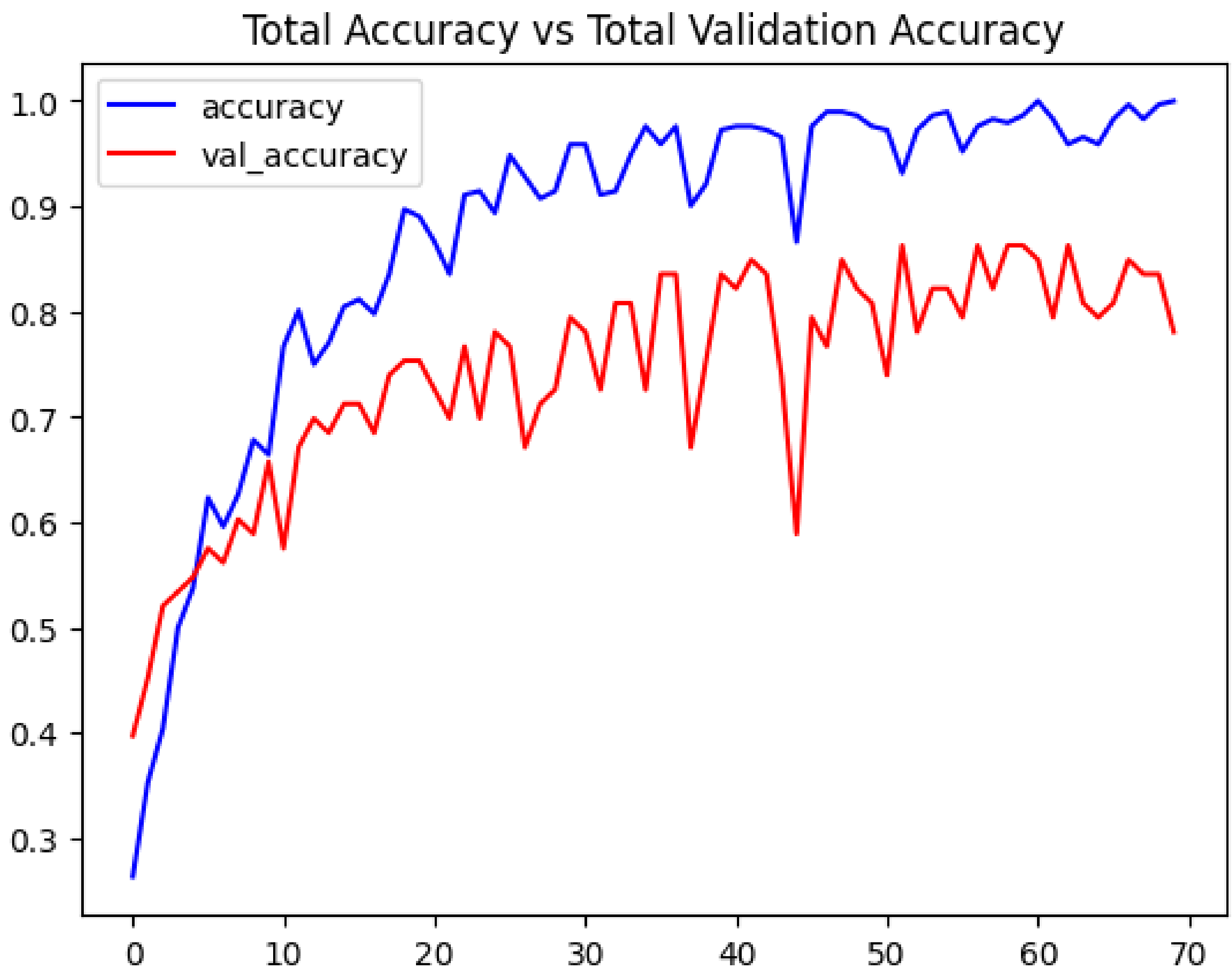
Strengths: excels on fluid motion (WalkingWithDog, HorseRace).

Limitations: confusion between visually similar motions (Swing vs TaiChi).

LRCN Results



LRCN Results



LRCN – Results

Test Accuracy: 86.93 %

Test Loss: 0.2854

Strengths: leverages strong frame-level spatial cues.

Very few misclassifications observed.

ConvLSTM vs LRCN

- Overall Accuracy: 79 % vs 87 %
- LRCN lower loss; better generalization.
- ConvLSTM better at continuous motion; LRCN excels with salient frames.
- Both models stable in training (early stopping).

Integration with HMM/DBN Pipeline

Recognized activities feed as observations to HMM/DBN for movement forecasting.

Creates proactive smart-home automation loop.

Demonstrates synergy between deep vision models and probabilistic predictors.

Key Takeaways

Deep spatiotemporal models achieve strong HAR on UCF50 subset.

LRCN outperforms ConvLSTM under identical training regime.

Findings guide model selection for real-time HAR applications.

Future Directions

SCALE TO FULL 50-CLASS UCF50 OR LARGER DATASETS (UCF101, HMDB51).

DATA AUGMENTATION & LONGER FRAME SEQUENCES.

INCORPORATE PRETRAINED CNNS AND ATTENTION/TRANSFORMER BLOCKS.

EDGE DEPLOYMENT VIA MODEL COMPRESSION (TENSORFLOW LITE).

References

- Donahue et al., 2015 – Long-term Recurrent Convolutional Networks.
- Shi et al., 2015 – ConvLSTM for precipitation nowcasting.
- Tran et al., 2015 – 3D CNNs for spatiotemporal learning.
- Simonyan & Zisserman, 2014 – Two-stream CNNs.
- Soomro et al., 2012 – UCF50 dataset introduction.

- Thank You