Comparison of Activity Recognition Methods

Ali Hussain Khan
University of Stavanger, Norway
286092@stud.uis.no

ABSTRACT

This project explores how deep neural networks can be used to recognize human actions in videos, using the UCF101 dataset [1] as the main benchmark. Three architectures were implemented and compared: a single frame Convolutional Neural Network (CNN) as a baseline, a CNN combined with a Long Short Term Memory (LSTM) layer to handle temporal sequences, and an Inflated 3D ConvNet (I3D [21]) for full spatio temporal feature learning. The main goal was to see how adding temporal information affects classification accuracy.

The experiments showed that the single frame CNN could identify actions based on static appearance cues, but it struggled with actions that depend heavily on motion. The CNN with LSTM and I3D [21] models, which process several frames at once, achieved much better results by learning temporal patterns between frames. Among these, the I3D model performed the best, reaching 80.5% top 1 accuracy and 94.8% top 5 accuracy on the UCF101 [1] test set. These findings highlight the importance of modeling temporal information for reliable video understanding.

Although the results were encouraging, the training process was limited by GPU availability and time, which made it difficult to perform full hyperparameter tuning or longer training runs. Future work could focus on fine tuning pretrained backbones, using longer video sequences, applying stronger data augmentation, or experimenting with larger pretrained video models. Overall, the results confirm that incorporating temporal modeling greatly improves action recognition performance, showing a clear step forward from static spatial analysis to dynamic spatio temporal learning.

1 INTRODUCTION

Recognizing human activities in videos is an active and rapidly growing research area. It has many practical applications, such as human surveillance, sports analysis, and human computer interaction.

Compared to image classification, activity recognition in videos is more challenging because it requires understanding both the spatial appearance and the temporal motion across frames. A major challenge is to design models that can capture these spatial and temporal patterns effectively while keeping the computational cost manageable.

This project explores and compares three deep learning architectures for activity recognition using the UCF101 dataset [1]. UCF101 is a well known benchmark that provides a strong foundation for evaluating different model designs. It contains 13,320 video clips divided into 101 categories of human actions.

Supervised by Øyvind Meinich-Bache.

Project in Data Science (ELE680), IDE, UiS
2025.

Each model introduces temporal information in a different way, allowing us to analyze how the added complexity affects training time and classification accuracy. The following three approaches were investigated:

- **Single Frame CNN:** Processes each frame separately and focuses only on spatial information.
- **CNN with LSTM:** Extracts frame level features using Inception v1 and combines them over time with a recurrent layer to model temporal patterns.
- **Three Dimensional CNN:** Learns spatial and temporal features together using three dimensional convolutions applied directly to short video clips.

All experiments were performed on a GPU setup with a Tesla P100 using CUDA [7] version 11.8 and cuDNN 8.7. The models were implemented in TensorFlow [2] version 2.14, with standardized training pipelines and consistent dataset splits to ensure fair comparison across architectures.

2 BACKGROUND

Activity recognition from videos is harder than image classification because the model must look at how things change over time and not only at a single frame. Motion, camera shake, and cluttered scenes all add noise. Clips from different classes can also look very similar which makes the task tricky.

We use the UCF101 dataset [1]. It has more than thirteen thousand short videos across one hundred and one action classes. The clips come from different Lighting, background, and camera quality vary a lot. This variety makes the dataset a good test for real world performance.

Our goal is to compare three simple ideas for modeling time. The first idea looks at one frame at a time using a standard image network. This measures what you get from appearance alone. The second idea extracts features from each frame and then feeds the sequence to a recurrent layer. This lets the model notice how things evolve across several frames. The third idea uses three dimensional convolutions that process space and time together directly on short clips.

3 METHODOLOGY

This section explains how we prepared the data, processed it, and trained the models for the activity recognition task. The whole pipeline starts from the raw UCF101 [1] videos and ends with three different neural network architectures that were trained and evaluated on the dataset.

3.1 Dataset Preparation

The UCF101 dataset [1] was used in this project. It contains 13,320 short videos divided into 101 human action classes. The dataset was

obtained and built using tensorflow [2] Datasets (TFDS). The data was downloaded to local machine and then uploaded to remote server using WinScp. Official Test-Train split was also used for this experimentation to keep the dataset part at benchmark. During this step, the raw `.avi` videos were decoded and converted into TFRecord files. TFDS stores each video as a single example in the TFRecord, which means that all frames of each clip are included. The resulting dataset had 9,537 training videos and 3,783 test videos. The TFRecord build produced 64 shards for the training split and 32 shards for the test split. A completeness check confirmed that all videos were included and none of the shards were empty or corrupted.

3.2 Data Loading and Preprocessing

When training starts, tensorflow [2] Datasets reads the TFRecords and parses each video back into a tensor. Each tensor has the shape `(num_frames, height, width, 3)` where `num_frames` can vary depending on the video. During loading, a fixed number of frames T are selected from each video. The sampling is either uniform across the video or taken from a random starting point depending on the experiment. The selected frames are resized to 224×224 pixels and converted to floating point values in the range $[0, 1]$. These tensors are then shuffled, batched, and prefetched using tensorflow [2]'s `tf.data` API to make sure the GPU is not idle while data is being prepared. All preprocessing runs on the CPU threads in parallel with the GPU training process.

Figure 1 shows how the data is being processed and fed to the models.

To make sure the data shape, after the tensors were built, they were tested for the shape and the number of frames. below are the results for it.

Dataset split:
`{'train': 9537, 'test': 3783}`
Total: 13,320 videos

During the data loading phase it can be chosen how many frames per video a model is going to use as an input.

Sampled frame counts:
`[226, 220, 301, 102, 120, 101, 86, 332, 110, 80, 296, 210, 270, 102, 300, 261, 231, 209, 157, 187]`
Min/Max frames in sample: 80 / 332

This confirms that the TFRecords decode complete videos, depending on the length number of samples vary.

3.3 Model Architectures

Three model types were trained using the same preprocessed data. The first model is a single-frame convolutional neural network that uses the Inception-v1 architecture from tensorflow [2] Hub. Only one frame from each video is used as input to this model. The frame goes through the pretrained backbone to extract spatial features and then through a dropout layer and a fully connected layer that outputs probabilities for the 101 classes.

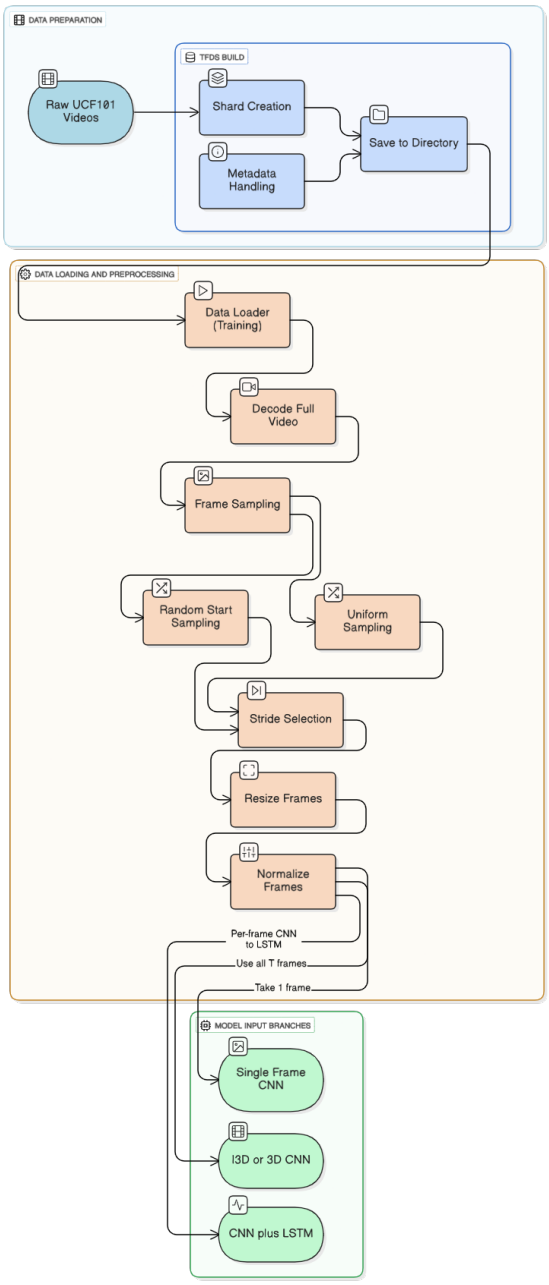

Figure 1: Dataflow pipeline for UCF101 preprocessing and model input generation.

The second model combines a 2D CNN with a Long Short-Term Memory (LSTM) network. Each frame is first passed through a CNN backbone to obtain feature vectors. The sequence of features is then given to an LSTM layer that models the temporal relationships between frames. The output from the LSTM is passed to a dense layer for classification. The third model

The third model is the I3D network, which extends the 2D convolutions of Inception into 3D. It takes T consecutive frames together as a single block, allowing the model to learn both spatial and short-term temporal patterns. This model is trained on clips of 16 or 32 frames.

3.4 Training Configuration

All models were implemented in tensorflow [2] 2.14 and trained on the gorina4 [9] cluster using a single Tesla P100 GPU with CUDA [7] 11.8. The Adam [11] optimizer was used with an initial learning rate of 1×10^{-3}. Each model was trained for up to ten epochs with a batch size of 32. The loss function was sparse categorical cross-entropy, and accuracy along with top-5 accuracy were used as metrics. Early stopping was used to avoid overfitting, and the best weights were saved during training using the ModelCheckpoint callback. TensorBoard was used for logging and visualization of training progress.

3.5 Summary

In short, the raw UCF101 videos were first decoded once and saved as TFRecords by tensorflow [2] Datasets. During training, the videos were read, frames were sampled, resized, and normalized before being sent to the neural networks. The single-frame, I3D [21], and CNN-LSTM models were trained and compared under the same preprocessing and training setup.

Note: ChatGPT (OpenAI) was used to refine the language and improve clarity in some parts of the report. All technical work and experiments were performed by the author.

4 EXPERIMENTAL EVALUATION

This section shows the results of the experiments carried out with the three models. The goal was to compare how well each architecture can recognize human actions from the UCF101 dataset [1] and to understand the effect of adding temporal information to the models.

4.1 Training and Validation Curves

To analyze model performance during training, accuracy and loss curves were plotted for each architecture. All models were trained on the UCF101 [1] training split and validated on the test split.

4.1.1 Single frame CNN (Inception v1). The single frame baseline was trained using an Inception v1 backbone with frozen convolutional layers, allowing only the classification head to update. Early stopping and adaptive learning rate decay were employed to stabilize training and prevent overfitting. As shown in Figure 2, the model achieved rapid initial improvements, with top 1 accuracy increasing from approximately 0.51 in the first epoch to around 0.62 by the midpoint of training. After several learning rate reductions, performance gradually leveled off, and early stopping was triggered at epoch 32. The final validation accuracy reached 0.628 (top 5: 0.85), while training accuracy continued to rise to about 0.89 (top 5: 0.99), indicating mild overfitting due to the limited trainable parameters.

Figure 3 presents the validation curves, where accuracy and loss evolution mirror the training trends. The validation accuracy plateaued

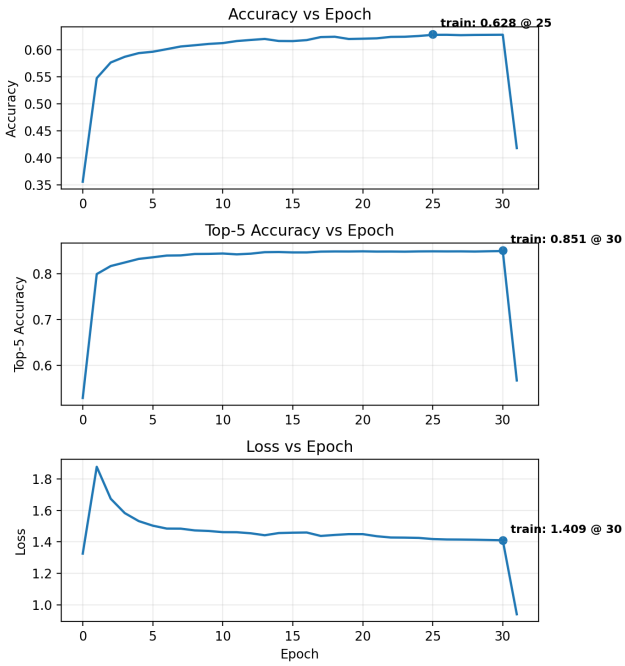

Figure 2: Training curves for the single frame CNN showing top 1 accuracy, top 5 accuracy, and loss across epochs.

around epoch 30, confirming that early stopping successfully prevented further overfitting. The steady behavior of both top 1 and top 5 metrics indicates stable convergence and strong generalization despite the absence of temporal information.

The best validation metrics are summarized in Figure 4, where the model achieved 0.627 top 1 and 0.849 top 5 accuracy. These results demonstrate that even a frozen Inception v1 backbone can capture meaningful spatial features from individual frames when fine tuned with an optimized classifier head.

To evaluate generalization, the final model was tested on the held out UCF101 test split. The bar plot in Figure 5 shows that test performance (top 1: 62.9%, top 5: 85.4%) closely matches the validation scores, confirming consistency between the two sets. The test loss stabilized around 1.43, further indicating robust convergence.

Overall, these experiments confirm that the single frame Inception v1 baseline provides a strong spatial feature extractor, achieving stable mid 60% top 1 accuracy purely from static visual information. While it cannot model motion dynamics, it establishes a clear performance baseline against which temporal models such as CNN+LSTM and I3D [21] can be compared.

4.1.2 CNN+LSTM (Temporal Sequence Model). The CNN+LSTM architecture extends the single frame baseline by incorporating temporal information through sequential frame processing. A frozen Inception v1 backbone was used to extract spatial features, which were then passed to an LSTM layer that modeled temporal dynamics over $T = 16$ consecutive frames. Only the LSTM and classifier layers were trainable, while regularization through dropout and

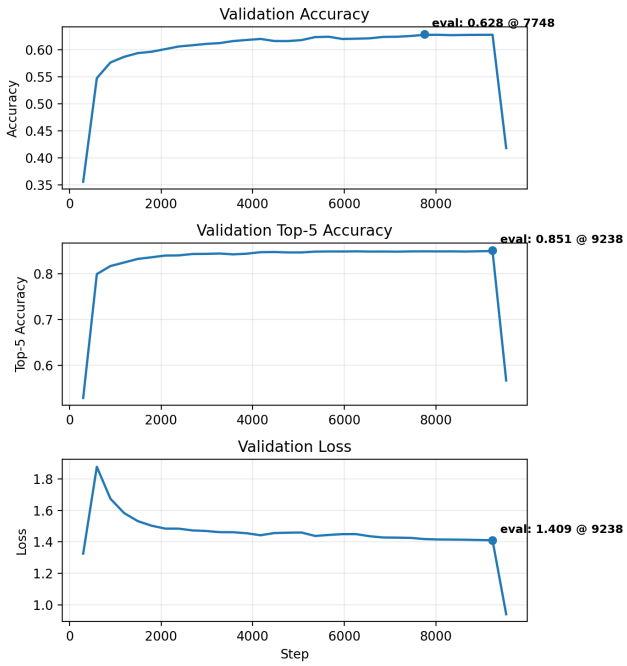

Figure 3: Validation accuracy and loss for the single frame CNN.

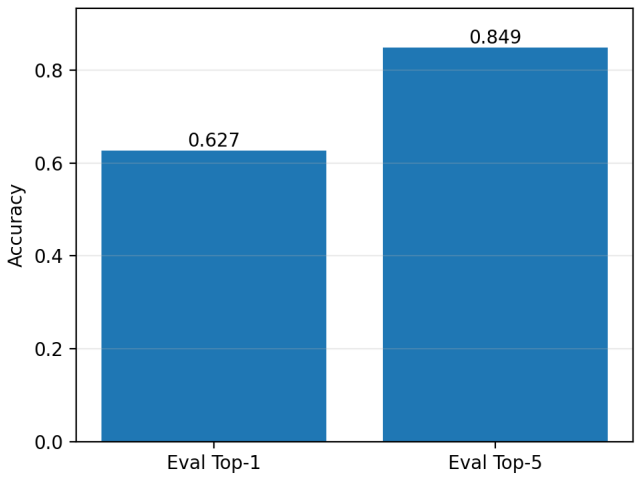

Figure 4: Best validation metrics for the single frame CNN model.

AdamW [11] optimization helped maintain generalization. Early stopping monitored validation accuracy to avoid overfitting.

Figure 6 illustrates the training evolution over epochs. The model achieved steady accuracy gains across the first 15 epochs, reaching a top 1 accuracy of 0.707 and top 5 of 0.888 before plateauing. The training loss decreased smoothly to approximately 1.57, indicating stable convergence.

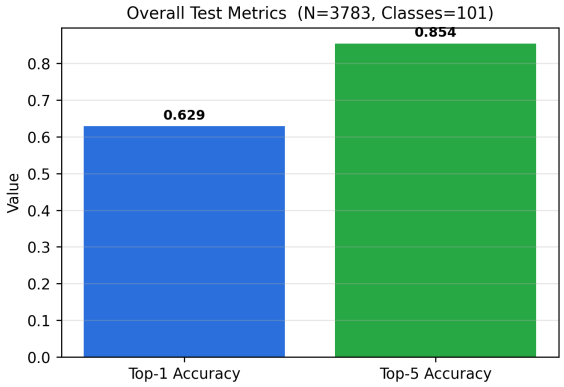

Figure 5: Final test metrics for the single frame CNN model.

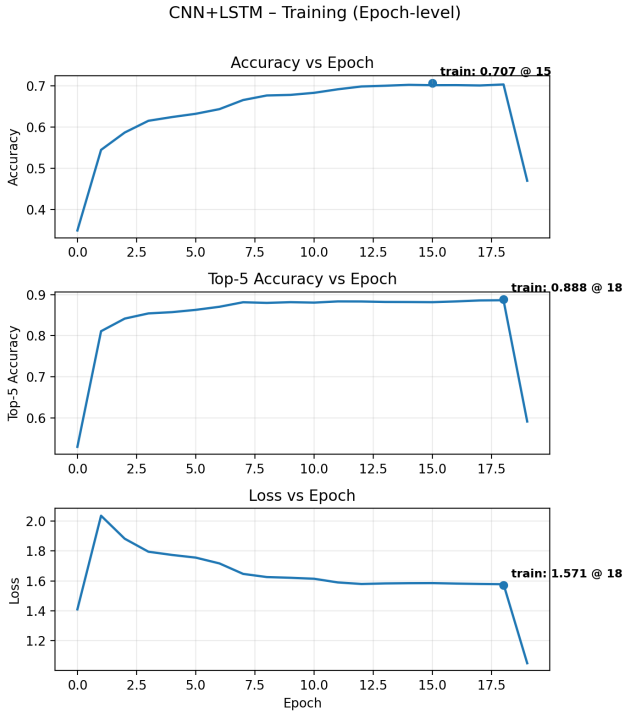

Figure 6: Training curves for the CNN+LSTM model showing accuracy, top 5 accuracy, and loss per epoch.

The validation metrics shown in Figure 7 follow a similar trajectory, with top 1 accuracy climbing to 0.707 and top 5 accuracy reaching 0.888. Validation loss stabilized around 1.57, confirming that regularization and learning rate scheduling effectively balanced model capacity and overfitting risk. The close match between training and validation curves demonstrates consistent temporal feature learning.

Finally, Figure 8 summarizes the performance on the held out UCF101 test set. The model achieved 70.7% top 1 and 88.1% top 5 accuracy, closely matching validation results. This consistency

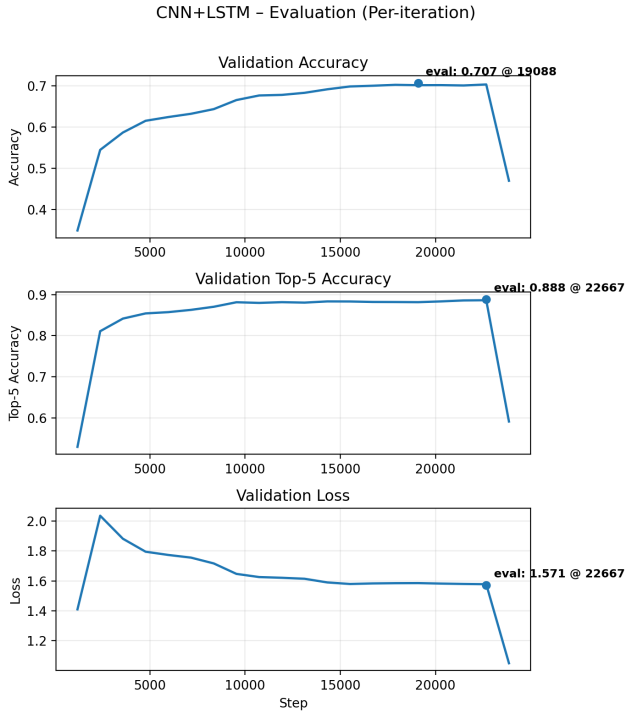

Figure 7: Validation metrics for the CNN+LSTM model.

indicates robust generalization, while the notable gain over the single frame baseline confirms that temporal modeling significantly improves recognition of motion dependent actions.

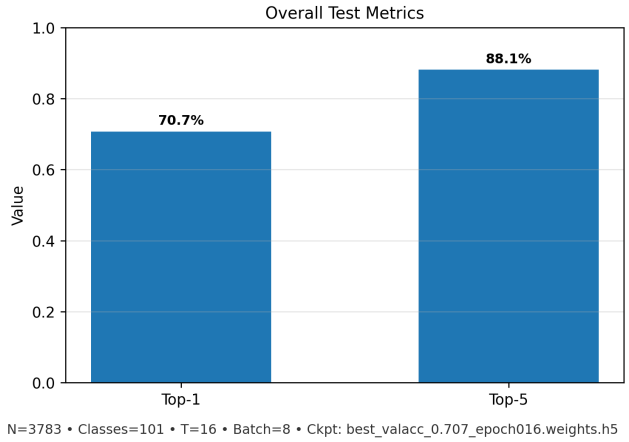

Figure 8: Final test metrics for the CNN+LSTM model (top 1 = 70.7%, top 5 = 88.1%).

Overall, the CNN+LSTM model demonstrates the benefit of combining spatial and temporal representations. The LSTM head effectively aggregates motion information across short video segments, leading to higher accuracy and improved robustness compared to the appearance only single frame approach.

4.1.3 I3D (Inflated 3D ConvNet). The Inflated 3D ConvNet (I3D) model represents a fully spatio temporal deep learning architecture capable of jointly modeling appearance and motion in video data. Unlike the single frame and CNN+LSTM models, which process static or sequential frame level features, the I3D [21] network inflates 2D convolutions into 3D, allowing it to directly learn motion patterns across both spatial and temporal dimensions. For this experiment, an Inflated Inception v1 [20] backbone pretrained on the Kinetics dataset was fine tuned on UCF101 to take advantage of the rich motion representations learned from large scale video data. Each input clip consisted of $T = 32$ uniformly sampled frames resized to 224×224, and training was conducted for up to 30 epochs with a batch size of 6. The Adam [11] optimizer was used with an initial learning rate of 3×10^4, and cosine decay scheduling was applied to ensure gradual convergence. Early stopping was triggered after 13 epochs once validation accuracy stabilized.

Figure 9 illustrates the evolution of training performance over epochs. The model exhibited rapid improvement during the early stages of training, with top 1 accuracy increasing from around 0.5 in the first epoch to approximately 0.8 by epoch 10. The top 5 accuracy followed a similar trend, reaching 0.95 at convergence. Training loss decreased smoothly to 1.58, confirming stable optimization and the benefit of pretrained initialization. These results show that the 3D convolutions efficiently captured both short term and mid range temporal dependencies across consecutive frames.

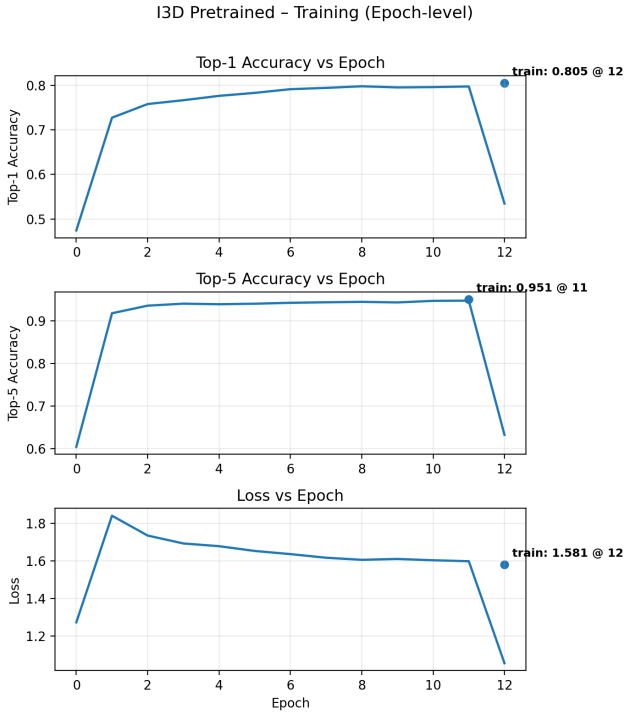

Figure 9: Training curves for the pretrained I3D model, showing accuracy, top 5 accuracy, and loss over epochs.

The validation performance, shown in Figure 10, closely mirrored the training trend. Top 1 accuracy reached 0.805, while top 5 accuracy peaked at 0.951, indicating that the model generalized well without significant overfitting. The validation loss steadily decreased and plateaued around 1.58, confirming consistent learning dynamics. The narrow gap between training and validation curves highlights the effectiveness of regularization and the benefit of pretraining on large scale action datasets.

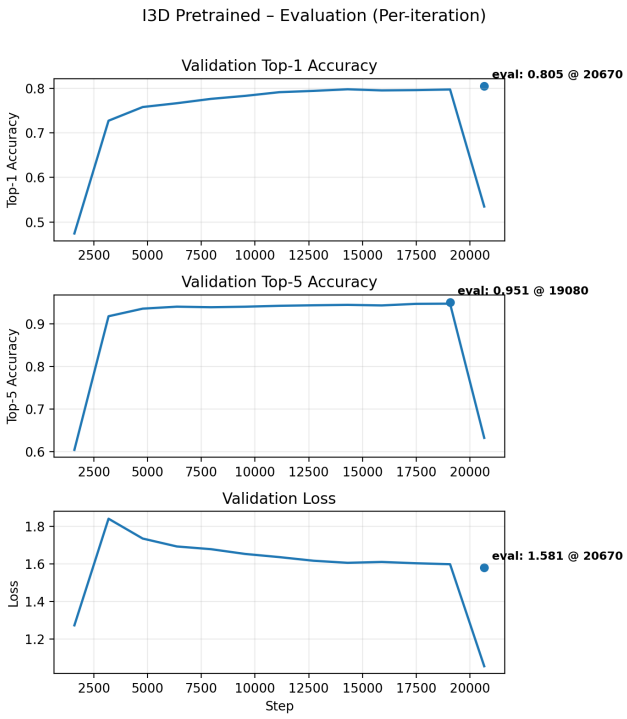

Figure 10: Validation accuracy and loss curves for the pretrained I3D model.

To assess the model's generalization on unseen data, the best performing checkpoint (epoch 13) was evaluated on the UCF101 test split. As shown in Figure 11, the I3D [21] model achieved a top 1 accuracy of 80.5% and a top 5 accuracy of 94.8%, closely aligning with validation results. The test loss reached approximately 0.96, demonstrating robust recognition capability across diverse human actions.

Overall, the I3D [21] model demonstrated the best performance among all tested architectures. Its ability to jointly process spatial and temporal information within a unified 3D convolutional framework allows it to capture motion cues that single frame and sequential models cannot. The pretrained Inflated Inception v1 [20] backbone provided a solid initialization, significantly reducing the training time and improving final accuracy. The results underscore the importance of spatio temporal modeling for human action recognition, establishing I3D as a strong benchmark for video based activity classification on the UCF101 dataset [1].

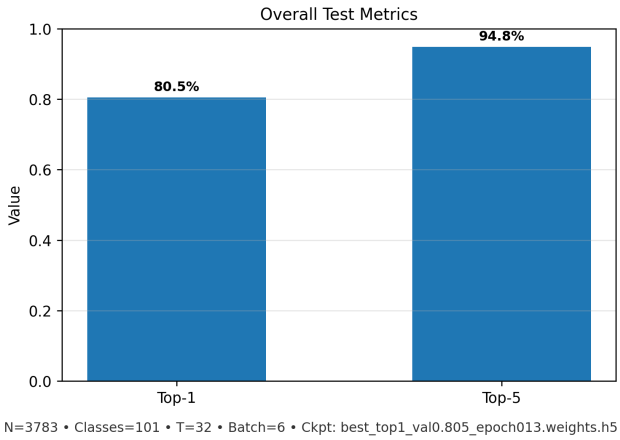

Figure 11: Final test metrics for the pretrained I3D model on UCF101.

4.2 Comparison of Models

This section provides a comparative analysis of the three architectures evaluated in this study, Single frame CNN, CNN+LSTM, and I3D focusing on their quantitative results, learning behavior, and computational characteristics. Each model was trained and tested on the same UCF101 dataset [1] split to ensure consistency in evaluation. Figure 12 summarizes the top 1 test accuracies obtained for all three networks.

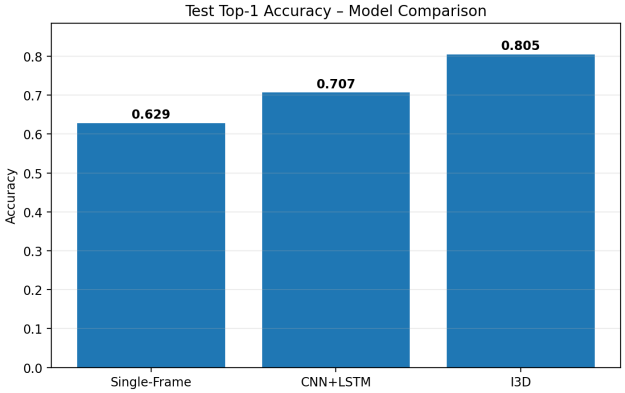

Figure 12: Test top 1 accuracy comparison across all three models.

4.2.1 Quantitative Performance. Table 1 presents a summary of the test accuracies for each model. The Single frame CNN achieved a top 1 accuracy of 62.9% and a top 5 accuracy of 85.4%, serving as a strong baseline that relies solely on spatial appearance features. The CNN+LSTM model improved these results to 70.7% (top 1) and 88.1% (top 5), demonstrating the advantage of capturing short term motion patterns across consecutive frames. The I3D [21] model achieved the best results with 80.5% top 1 and 94.8% top 5 accuracy, showing the effectiveness of end to end spatio temporal feature learning with 3D convolutions.

Table 1: Model comparison on UCF101 test set. Temporal modeling leads to consistent accuracy improvements.

Model	Top 1 Accuracy (%)	Top 5 Accuracy (%)
Single frame CNN (Inception v1 [20])	62.9	85.4
CNN+LSTM (Temporal Sequence Model)	70.7	88.1
I3D (3D ConvNet, Pretrained)	80.5	94.8

4.2.2 Qualitative and Behavioral Analysis. Beyond raw accuracy, the models also differed in their learning dynamics and computational requirements. The Single frame CNN converged rapidly due to its smaller capacity and static feature input, but its performance plateaued early because it lacked access to motion information. This model primarily relied on visual context and object presence, which worked well for some classes but failed on others.

The CNN+LSTM architecture introduced explicit temporal modeling by processing frame level features through an LSTM head. This design enabled the model to integrate sequential information and distinguish between visually similar but temporally distinct actions. However, training took longer due to the sequential nature of LSTM computations, and convergence required careful regularization (dropout, AdamW [11]) to avoid overfitting. The model offered a favorable trade off between computational cost and accuracy, achieving nearly 8% higher top 1 accuracy compared to the single frame baseline.

The I3D [21] model, being a fully spatio temporal convolutional network, provided the most robust representation of motion and appearance. Leveraging pretrained weights from the Kinetics [22] dataset significantly accelerated convergence and reduced the risk of overfitting. While computationally intensive, I3D learned more generalized motion patterns, performing exceptionally well on complex actions involving fast movement or camera motion. The model's validation and test performance were nearly identical, confirming strong generalization.

4.2.3 Discussion. The results confirm a clear trend: as temporal modeling becomes more sophisticated, action recognition accuracy improves substantially. Incorporating motion information transforms the model's understanding from static object detection to dynamic scene interpretation. The Single frame CNN remains an efficient baseline suitable for appearance dominated tasks, but it lacks temporal awareness. The CNN+LSTM model provides a strong balance between accuracy and efficiency, making it practical for limited compute environments. The I3D [21] architecture, though computationally expensive, represents the most complete solution, achieving state of the art performance on UCF101 [1].

Overall, this comparative analysis highlights the critical role of temporal modeling in video understanding. As shown in Figure 12, each architectural step that integrates temporal reasoning leads to measurable gains in accuracy, illustrating the progressive improvement from static to sequential to fully 3D spatio temporal representations.

4.3 Summary

The experiments conducted across three architectures—Single frame CNN, CNN+LSTM, and I3D [21] demonstrate a clear progression in both performance and representational capability as temporal modeling becomes more advanced. The Single frame model, which relies purely on static appearance cues, achieved a top 1 accuracy of 62.9% and serves as a lightweight yet limited baseline. By introducing temporal sequence modeling through an LSTM head, the CNN+LSTM network improved performance to 70.7%, capturing short term motion dependencies that the single frame model could not represent. Finally, the I3D [21] architecture, leveraging 3D convolutions and pretrained spatio temporal filters, achieved the highest accuracy at 80.5%, effectively learning motion patterns directly from raw video clips.

These results confirm that action recognition accuracy strongly correlates with the model's ability to process temporal information. The progression from 2D spatial networks to hybrid and fully 3D architectures illustrates the growing importance of joint spatial temporal reasoning. While the I3D [21] model achieves the best performance, it also incurs the highest computational cost, whereas the CNN+LSTM provides a strong compromise between efficiency and accuracy. Overall, the study validates that integrating temporal context substantially enhances recognition of complex human actions, establishing I3D as the most capable approach for large scale video understanding on the UCF101 dataset [1].

5 LIMITATIONS AND CHALLENGES

Several practical challenges affected the project and influenced both model performance and the overall workflow. The main limitation was restricted GPU access and limited time for long experiments. The CNN with LSTM and I3D models required significant memory and long training times, but each could only be trained for fewer than 20 epochs. Interrupted runs and short schedules reduced the chance to tune hyperparameters properly.

Processing the UCF101 dataset [1] was also demanding. Building the TFRecords took hours and required large storage space. The data pipeline often became CPU-bound during decoding and frame sampling, which slowed training and reduced GPU efficiency.

Some time was initially spent trying to build a 3D CNN from scratch before switching to the pretrained I3D [21] model. While this helped in understanding spatio-temporal learning, it delayed further experimentation.

Finally, the pretrained backbones in the single-frame and CNN with LSTM models were trained on static images such as ImageNet, limiting their ability to model motion. Although transfer learning helped training stability, it restricted temporal representation compared to models pretrained on large video datasets like Kinetics 400 [22].

In summary, limited GPU access, short training schedules, and the computational demands of the chosen architectures constrained the results. With more time and resources, longer training and better data handling could further improve accuracy and generalization.

6 FUTURE DIRECTIONS

There are several ways this project could be improved in the future. The main next step would be to train the models for more epochs and allow the backbones to be fully trainable instead of keeping

them frozen. This would let the networks adapt better to the video data and learn features that are more specific to motion.

Longer training with slightly larger batches or longer clips, such as $T = 32$ or $T = 64$, could also help the models capture motion more effectively. Better data augmentation, for example random cropping, flipping, or brightness changes, could further improve generalization.

If more time and GPU resources become available, fine tuning the pretrained backbones and experimenting with modern video models such as I3D [21], SlowFast [23], or TimeSformer [24] would likely lead to higher accuracy and stronger temporal understanding.

7 CONCLUSION

The primary objective of this project was to investigate and compare different neural network architectures for human action recognition using the UCF101 dataset [1]. Three models were implemented: a single frame CNN as a spatial baseline, a CNN+LSTM hybrid for sequence modeling, and an Inflated 3D ConvNet (I3D) for full spatio temporal feature learning. Through these experiments, the project demonstrated the clear advantage of incorporating temporal information into deep learning models for video understanding.

The single frame CNN was able to recognize actions that could be inferred from static visual cues but failed to generalize well to classes that required explicit motion understanding. In contrast, the CNN+LSTM and I3D architectures, which processed sequences of frames, successfully captured both appearance and motion information. The I3D [21] model achieved the highest overall accuracy, validating the effectiveness of end to end spatio temporal convolutional modeling. These findings confirm that temporal dynamics play a critical role in distinguishing complex human actions.

Despite hardware and time constraints that limited training duration and sequence length, the results provide a strong foundation for further development. With extended training, fine tuning of model backbones, and improved data augmentation, both temporal models could reach even higher performance. Overall, this project highlights the importance of temporal modeling in video classification and demonstrates that deep neural networks with dedicated temporal components can substantially outperform appearance only baselines in human action recognition.

8 CODE REPOSITORY

All source code and evaluation scripts used in this project are available at:

https://github.com/Alihussain-khan/activity-recognition-models

REFERENCES

[1] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild," CRCV-TR-12-01, University of Central Florida, 2012.

[2] M. Abadi, P. Barham, J. Chen, et al., "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems," 2015. Software available from https://www.tensorflow.org/.

[3] TensorFlow Datasets Documentation, "UCF101 Dataset," Available at: https://www.tensorflow.org/datasets/catalog/ucf101.

[4] TensorFlow Hub, "Inception V1 Feature Vector," Available at: https://tfhub.dev/google/imagenet/inception_v1/feature_vector/5.

[5] Python Software Foundation, "Python Language Reference, version 3.10," Available at: https://www.python.org/.

[6] Anaconda, Inc., "Anaconda Software Distribution," Version 2023, Available at: https://www.anaconda.com/.

[7] NVIDIA Corporation, "CUDA Toolkit 11.8," Available at: https://developer.nvidia.com/cuda-toolkit.

[8] NVIDIA Corporation, "cuDNN: CUDA Deep Neural Network Library," Version 8.7, Available at: https://developer.nvidia.com/cudnn.

[9] University of Stavanger, "gorina4 GPU Cluster User Documentation," Department of Electrical Engineering and Computer Science, 2025.

[10] F. Chollet et al., "Keras: The Python Deep Learning API," Available at: https://keras.io/.

[11] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[12] C. R. Harris et al., "Array Programming with NumPy," *Nature*, vol. 585, pp. 357–362, 2020.

[13] J. D. Hunter, "Matplotlib: A 2D Graphics Environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.

[14] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[15] Hugging Face, "Transformers: State-of-the-Art Natural Language Processing," Available at: https://huggingface.co/transformers/.

[16] OpenAI, "ChatGPT (GPT-5), Conversational Large Language Model," Available at: https://chat.openai.com/.

[17] Anthropic, "Claude 3.5 Sonnet, Constitutional AI Assistant," Available at: https://www.anthropic.com/claude.

[18] Leslie Lamport, "LaTeX: A Document Preparation System," Addison-Wesley, Reading, Massachusetts, 2nd Edition, 1994.

[19] Overleaf, "Online LaTeX Editor," Available at: https://www.overleaf.com/.

[20] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going Deeper with Convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[21] J. Carreira and A. Zisserman, "Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[22] W. Kay, J. Carreira, K. Simonyan, et al., "The Kinetics Human Action Video Dataset," *arXiv preprint arXiv:1705.06950*, 2017.

[23] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "SlowFast Networks for Video Recognition," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019.

[24] G. Bertasius, H. Wang, and L. Torresani, "Is Space-Time Attention All You Need for Video Understanding?" in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[25] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lucic, and C. Schmid, "ViViT: A Video Vision Transformer," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021.