2026 Volume: 02 Issue: 04 www.ijrpa.com ISSN 2456-9995 Review Article

International Journal Research Publication Analysis

Page: 01-12

VIDEO ACTION RECOGNITION: A REVIEW OF RESNET-LSTM AND ATTENTION-ENHANCED RESNET ARCHITECTURES

[*1]Dr. Ramya. B.N., [2]Yashwanth. B. L., [3]Sumanth Gowda. B. S.

[1]Associate Professor, Department of Computer Science and Engineering, Jyothy Institute of Technology, Bengaluru, India.

[2,3]Department of Computer Science and Engineering, Jyothy Institute of Technology, Bengaluru, India.

Article Received: 18 March 2026
Article Revised: 08 April 2026
Published on: 28 April 2026

***Corresponding Author: Dr. Ramya. B.N.**
Associate Professor, Department of Computer Science and Engineering, Jyothy Institute of Technology, Bengaluru, India. DOI: https://doi-doi.org/101555/ijrpa.3751

ABSTRACT

Video action recognition has been revolutionising various sectors like security, healthcare, entertainment and sports analytics. Recent advances in deep learning have significantly improved the ability to model both spatial and temporal characteristics of a video. This paper presents two approaches for action recognition : 1.) The ResNet-LSTM; 2.) ResNet with Transformer encoder. We analyze the architectural design, performance characteristics, and limitations of both methods, highlighting their effectiveness in handling complex action sequences. This study aims to provide insights into the strengths of hybrid deep learning models and the role of attention mechanisms in advancing video understanding tasks. Code available at https://github.com/theoptiplexcoder/videoRecognition-lstm-transformers

KEYWORDS: Video Action Recognition, ResNet, LSTM, Transformer Encoder, Attention Mech- anisms, Sinusoidal positional embedding.

INTRODUCTION

Video has been popular form of data on the internet that people interact with due to advancements in technologies like internet, storage capacity, cameras, audio etc. Most of the videos are searched based on the keywords or categories but most of the videos on the internet are unlabelled or not categorized. This is the reason why there is exists a necessity to automatically classify videos based on the video content. Each day billions of videos are shared on social media platforms such as YouTube,X,Instagram, Facebook etc. presenting a requirement for

semi-supervised learning methods to classify videos based on the activity.

In this paper, we use Resnet-LSTM[7] architecture for video activity recognition where the ResNets capture the spatial features of a video whereas the Long Short Term Memory(LSTM) captures the temporal features of the video from the spatial features of each frame and then classified.

The other method is ResNet-Transformer[5] architecture with self-attention mechanisms, naturally capturing long-range dependencies allowing each token to attend all other tokens in other frames thus resulting in enhanced understanding of the activity context. This capability enables transformers to encode meaningful context information into video representations, facilitating a deeper understanding of the temporal dynamics and interactions within the video sequence.

Recent Work

Deep Learning for Video and Action Recognition Over the last decade, active research has shifted from hand-crafted features to deep learning methods to improve video classification and human action recognition Convolutional Neural Networks (CNNs) have achieved break- throughs in image processing by effectively capturing spatial features and local semantics.However, traditional 2D CNNs inevitably lose temporal information when mapping 3D sequences to 2D images.To capture spatio-temporal information more effectively, researchers have devel- oped 3D CNN architectures that extract features from both spatial and temporal dimensions, as well as two-stream CNNs that process single frames and multi-frame optical flow con- currently.Furthermore, Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks have been increasingly adopted because of their powerful ability to model long-term contextual dependencies within complex temporal sequences.[6] Hybrid Architectures To maximize the extraction of both spatial and temporal dynamics, recent works frequently employ hybrid architectures. Combining CNNs and LSTMs allows systems to leverage the spatial feature-extraction strength of CNNs alongside the sequential modeling capabilities of LSTMs. For example, in 3D skeleton-based action recognition, feeding spatial domain features into LSTMs and temporal domain features into CNNs, followed by late score fusion, has demonstrated state-of-the-art performance by compensating for the weak ability of RNNs to distinguish useful information from multiple feature types. Other approaches have successfully combined pre-trained CNNs with ConvLSTMs to achieve higher classification accuracy in video summarization and scene classification.[1] Transformers and Semi-Supervised Learning More recently, Vision

Transformers (ViT) have been adapted for video understanding, utilizing self-attention mechanisms that natively capture long-range temporal dependencies and complex interactions across distant frames. Advanced frameworks, such as ActNetFormer, propose a cross-architecture strategy that hybridizes 3D CNNs (which excel at local spatial-temporal features) with Video Transformers (which excel at global contextual understanding) to capture a more holistic view of the action.[5]

METHODOLOGY

1.1 Dataset

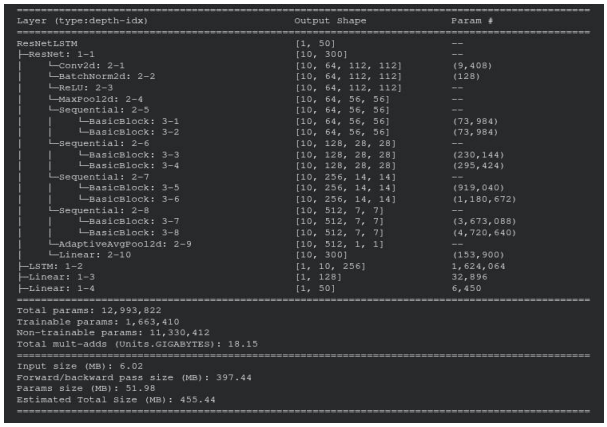
UCF50 is a benchmark action recognition dataset designed to emphasize real-world variability rather than controlled, staged scenarios.It is a collection of 50 action classes sourced directly from YouTube. UCF50 data set's 50 action categories collected from youtube are: Baseball Pitch, Basketball Shooting, Bench Press, Biking, Biking, Billiards Shot,Breaststroke, Clean and Jerk, Diving, Drumming, Fencing, Golf Swing, Playing Guitar, High Jump, Horse Race, Horse Riding, Hula Hoop, Javelin Throw, Juggling Balls, Jump Rope, Jumping Jack, Kayaking, Lunges, Military Parade, Mixing Batter, Nun chucks, Playing Piano, Pizza Tossing, Pole Vault, Pommel Horse, Pull Ups, Punch, Push Ups, Rock Climbing Indoor, Rope Climbing, Rowing, Salsa Spins, Skate Boarding, Skiing, Skijet, Soccer Juggling, Swing, Playing Tabla, TaiChi, Tennis Swing, Trampoline Jumping, Playing Violin, Volleyball Spiking, Walking with a dog, and Yo Yo.

1.2 Frame Extraction

A video capture object iteratively reads frames until either the video ends or a specified maximum number of frames is reached. Each frame undergoes a series of transformations then resized to a fixed spatial resolution (default 224×224 pixels), and finally converted from BGR to RGB color format by reordering the channel indices. The processed frames are accumulated in a list and, once extraction is complete, converted into a NumPy array for efficient numerical handling.

1.3 Model architecture

1.3.1 ResNet-LSTM



```
==========================================================================================
Layer (type:depth-idx)                             Output Shape              Param #
==========================================================================================
ResNetLSTM                                         [1, 50]                   --
├─ResNet: 1-1                                      [10, 300]                 --
│    └─Conv2d: 2-1                                 [10, 64, 112, 112]        (9,408)
│    └─BatchNorm2d: 2-2                            [10, 64, 112, 112]        (128)
│    └─ReLU: 2-3                                   [10, 64, 112, 112]        --
│    └─MaxPool2d: 2-4                              [10, 64, 56, 56]          --
│    └─Sequential: 2-5                             [10, 64, 56, 56]          --
│    │    └─BasicBlock: 3-1                        [10, 64, 56, 56]          (73,984)
│    │    └─BasicBlock: 3-2                        [10, 64, 56, 56]          (73,984)
│    └─Sequential: 2-6                             [10, 128, 28, 28]         --
│    │    └─BasicBlock: 3-3                        [10, 128, 28, 28]         (230,144)
│    │    └─BasicBlock: 3-4                        [10, 128, 28, 28]         (295,424)
│    └─Sequential: 2-7                             [10, 256, 14, 14]         --
│    │    └─BasicBlock: 3-5                        [10, 256, 14, 14]         (919,040)
│    │    └─BasicBlock: 3-6                        [10, 256, 14, 14]         (1,180,672)
│    └─Sequential: 2-8                             [10, 512, 7, 7]           --
│    │    └─BasicBlock: 3-7                        [10, 512, 7, 7]           (3,673,088)
│    │    └─BasicBlock: 3-8                        [10, 512, 7, 7]           (4,720,640)
│    └─AdaptiveAvgPool2d: 2-9                      [10, 512, 1, 1]           --
│    └─Linear: 2-10                                [10, 300]                 (153,900)
├─LSTM: 1-2                                        [1, 10, 256]              1,624,064
├─Linear: 1-3                                      [1, 128]                  32,896
├─Linear: 1-4                                      [1, 50]                   6,450
==========================================================================================
Total params: 12,993,822
Trainable params: 1,663,410
Non-trainable params: 11,330,412
Total mult-adds (Units.GIGABYTES): 18.15
==========================================================================================
Input size (MB): 6.02
Forward/backward pass size (MB): 397.44
Params size (MB): 51.98
Estimated Total Size (MB): 455.44
==========================================================================================
```

Figure 1: ResNet-LSTM model architecture.

The ResNet–LSTM architecture represents a hybrid deep learning framework designed to effectively model both spatial and temporal characteristics in sequential visual data, such as videos. In this architecture, a Convolutional Neural Network (CNN), typically a pre-trained Residual Network (ResNet), serves as the spatial feature extractor. Individual frames from a video sequence are passed through the ResNet, which leverages residual learning and deep convolutional layers to generate high-level feature embeddings that capture semantic and structural information within each frame.

To improve computational efficiency and prevent overfitting, the ResNet backbone is often frozen, and its final fully connected layer is adapted to produce compact feature vectors suitable for sequential processing. These frame-level features are then reshaped into a temporal sequence and fed into a Long Short-Term Memory (LSTM) network, which is specifically designed to model sequential dependencies through its gated architecture, including input, forget, and output gates. The LSTM processes the ordered feature vectors to learn temporal dynamics and long-range dependencies across frames, making it particularly effective for tasks such as action recognition. Regularization techniques such as dropout are applied within the LSTM and subsequent fully connected layers to enhance generalization. Finally, the temporally aggregated representation is passed through one or more dense layers, and a Softmax activation function is used to produce class probabilities for the target categories. This combined architecture ef- fectively integrates spatial perception with temporal reasoning, enabling robust performance in video-based classification tasks.

1.3.2 ResNet-Transformer

The ResNet–Transformer architecture, commonly referred to as ActNetFormer in recent literature, represents a hybrid deep learning framework that integrates convolutional feature ex-

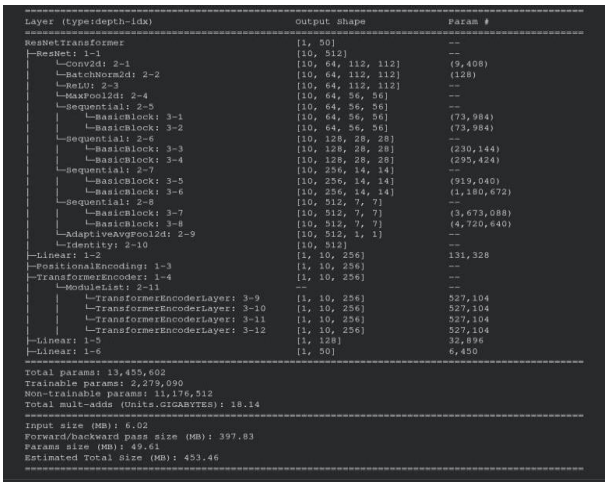


Figure 2: ResNet-transformer model architecture.

traction with attention-based sequence modeling for robust video understanding. In this architecture, a Residual Network (ResNet) serves as the primary backbone for spatial feature extraction, where individual video frames are processed independently to produce high-level feature embeddings that capture local semantics and appearance-based information. The final fully connected layer of the ResNet is typically removed or replaced with an identity mapping, allowing the network to output compact feature vectors (e.g., 512-dimensional), while freezing its parameters can stabilize training and reduce computational cost. These frame-wise features are subsequently projected into a lower-dimensional embedding space and enriched with positional encodings to retain temporal order information. The resulting sequence is then passed to a Transformer encoder, which employs multi-head self-attention mechanisms to model long-range temporal dependencies and global contextual relationships across the entire video sequence. Unlike recurrent architectures, the Transformer enables parallel processing and more effective learning of complex inter-frame interactions. To handle variable-length sequences, padding masks are incorporated during attention computation, ensuring that irrelevant positions do not influence the learned representations. The encoded sequence is then aggregated—either via masked average pooling or global mean pooling—to obtain a fixed-length video-level rep- resentation. Finally, this representation is passed through fully connected layers with non-linear activations to perform classification, typically followed by a Softmax function to produce class probabilities. This hybrid design effectively combines the locality-sensitive strengths of CNNs with the global reasoning capabilities of Transformers,

resulting in a powerful and scalable architecture for video action recognition tasks.

1.4 Workflow

The raw videos from the UCF50 dataset are first downloaded and extracted. The videos are then processed to sample frames and resized them to a fixed spatial resolution 224 x 224. A custom dataset class is defined to load a fixed number of frames per video along with applying data augmentations like normalization based on the ImageNet stats, horizontal flipping and colour jittering. Depending upon the model architecture the input frames are then processed and classification output is obtained. In our experiments we chose to use 5 consecutive frames per video for training due to computation constraints. We use Adam Optimizer with learning rate 3e-3, StepLR with step-size=3 and gamma=0.0005, EarlyStopper with patience=5. We run the training process for 30 EPOCHS and the final results are plotted.

RESULTS

1.5 Comparative analysis of Accuracy and Performance

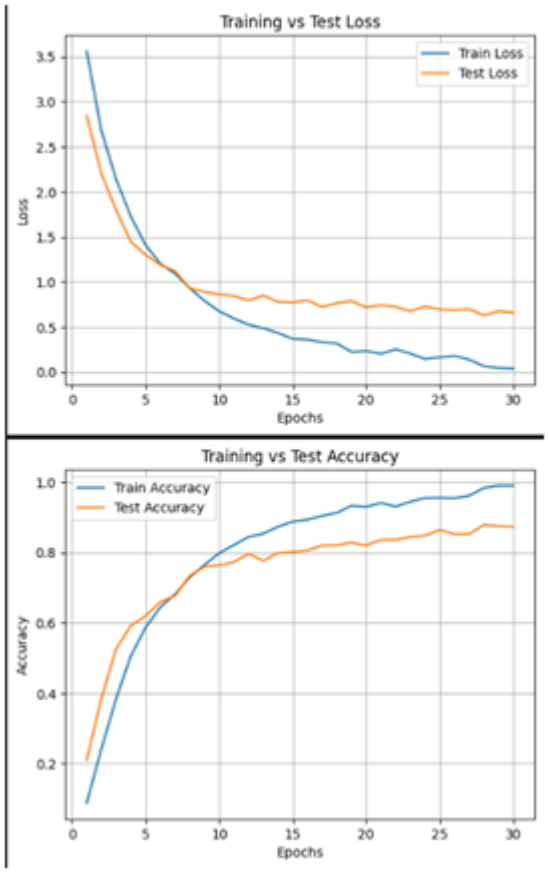


Figure 3: ResNet-LSTM performance graph.

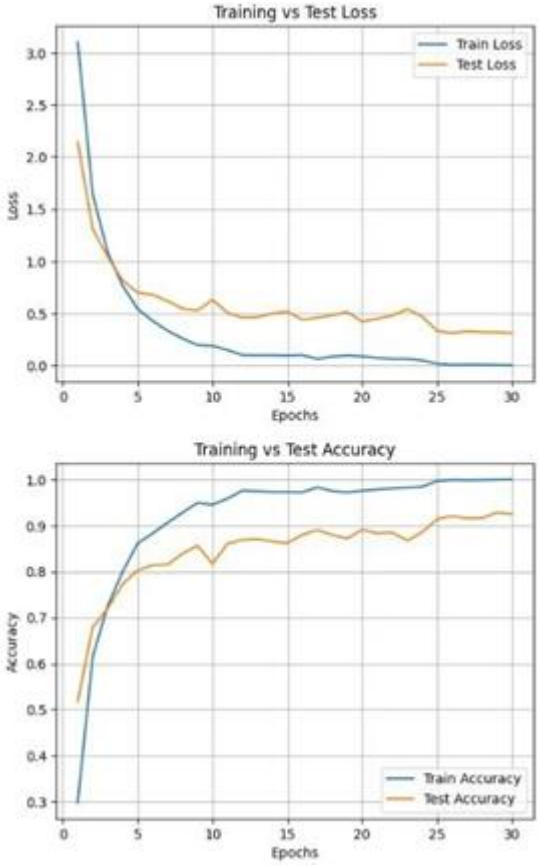


Figure 4:ResNet-Transformer perfor- mance graph.

The ResNet–Transformer model exhibits superior training dynamics compared to the ResNet–LSTM architecture. It demonstrates faster convergence during the initial training epochs, followed by stable optimization behavior. At 30 epochs, the model achieves a lower validation loss and a higher test accuracy of approximately 92%. Additionally, the smaller gap between training and validation performance indicates improved generalization and reduced overfitting.(see Fig. 4) In contrast, the ResNet–LSTM model shows a more gradual decrease in loss, with convergence plateauing around epochs 30. Furthermore, a noticeable gap between training and validation loss curves suggests the presence of mild overfitting, indicating that the model is less effective in generalizing to unseen data compared to the Transformer-based approach.(see Fig. 3) The performance of the proposed models is evaluated on a 50-class video classification task using a dataset of 6,671 samples, with an 80:20 train–test split. Tables (see Fig. 1) and (see Fig. 2)present the detailed classification reports for the ResNet-LSTM and ResNet-Transformer models, respectively. At an aggregate level, the ResNet-Transformer model achieves a classification accuracy of 92.52%, a notable increase in the ResNet-LSTM model, which attains 87.28% accuracy. This

improvement is consistently reflected across both macro-averaged F1-score (0.9226 vs. 0.8683) and weighted F1-score (0.9242 vs. 0.8703), indicating that the Transformer-based architecture not only improves overall performance but also maintains better balance across class distributions.

A deeper class-wise analysis reveals that the ResNet-Transformer model provides more stable and consistent predictions across the majority of classes. In particular, several classes that exhibited near-perfect performance under the Transformer model (e.g., classes of 1.0) demonstrate the model's ability to learn highly discriminative spatiotemporal representations.

In contrast, the ResNet-LSTM model shows noticeable variability in performance across classes. While certain classes achieve high F1-scores, a subset of classes suffers from significantly lower recall and F1-score values. Notably, classes 11, 45, and 48 exhibit poor performance, with F1- scores of 0.5366, 0.5128, and 0.6415, respectively. These results suggest that the LSTM-based temporal modeling struggles to capture discriminative temporal dependencies for these cate- gories.

Furthermore, the Transformer model significantly improves recall across multiple classes, suggesting enhanced sensitivity to class-specific features. Classes such as 6, 7, 8, 18, and 31 achieve recall values close to or equal to 1.0, highlighting the effectiveness of the attention mechanism in capturing long-range temporal dependencies across video frames.

Another important observation is the reduction in performance variance across classes. The ResNet-LSTM model exhibits a wider spread in F1-scores, whereas the ResNet-Transformer model maintains consistently high performance across most categories. This indicates improved generalization and robustness.

Despite the overall improvements, a few classes (e.g., class 38) show marginal gains, suggesting that attention mechanisms alone may not fully address challenges arising from ambiguous class boundaries or insufficient data representation.

Table 1: Classification Report for ResNet-LSTM.

Class	Precision	Recall	F1-score	Support
0	0.9062	0.9667	0.9355	30
1	0.8333	0.9259	0.8772	27
2	0.9143	1.0000	0.9552	32
3	0.8000	0.8276	0.8136	29
4	1.0000	1.0000	1.0000	30
5	0.9048	0.9500	0.9268	20
6	0.9091	0.9091	0.9091	22
7	0.9091	0.9677	0.9375	31
8	0.9355	0.9062	0.9206	32
9	0.8800	1.0000	0.9362	22
10	0.9200	0.8214	0.8679	28
11	0.6875	0.4400	0.5366	25
12	0.9583	0.9200	0.9388	25
13	0.9268	0.9500	0.9383	40
14	0.8462	0.8800	0.8627	25
15	0.6923	0.7826	0.7347	23
16	0.8846	0.9583	0.9200	24
17	0.9000	0.9000	0.9000	30
18	0.9200	0.9200	0.9200	25
19	0.8929	0.7812	0.8333	32
20	0.8800	0.7857	0.8302	28
21	0.9545	0.8400	0.8936	25
22	0.9655	1.0000	0.9825	28
23	0.8438	0.9000	0.8710	30
24	0.7692	0.8696	0.8163	23
25	1.0000	1.0000	1.0000	32
26	1.0000	0.9524	0.9756	21
27	1.0000	1.0000	1.0000	25
28	1.0000	1.0000	1.0000	20
29	0.7222	0.8125	0.7647	32
30	0.9167	0.8800	0.8980	25
31	0.8519	0.9583	0.9020	24
32	0.9688	0.9688	0.9688	32
33	1.0000	0.9048	0.9500	21
34	1.0000	0.7000	0.8235	30
35	0.7000	0.8077	0.7500	26
36	0.9259	0.9259	0.9259	27
37	0.8571	0.8889	0.8727	27
38	0.7200	0.7500	0.7347	24
39	0.9583	0.7931	0.8679	29
40	0.7692	1.0000	0.8696	20
41	0.8571	0.7742	0.8136	31
42	0.7429	0.9630	0.8387	27
43	0.8421	0.8000	0.8205	20
44	0.9697	0.9412	0.9552	34
45	0.7692	0.3846	0.5128	26

46	0.7200	0.7500	0.7347	24
47	0.8214	1.0000	0.9020	23
48	0.6071	0.6800	0.6415	25
49	0.9091	0.7692	0.8333	26
Accuracy		0.8728		1337
Macro Avg	0.8733	0.8721	0.8683	1337
Weighted Avg	0.8765	0.8728	0.8703	1337

Table 2: Classification Report for ResNet-Transformer.

Class	Precision	Recall	F1-score	Support
0	0.9355	0.9667	0.9508	30
1	0.8889	0.8889	0.8889	27
2	0.9412	1.0000	0.9697	32
3	0.9615	0.8621	0.9091	29
4	1.0000	1.0000	1.0000	30
5	0.9000	0.9000	0.9000	20
6	0.9565	1.0000	0.9778	22
7	0.9394	1.0000	0.9688	31
8	0.9697	1.0000	0.9846	32
9	1.0000	1.0000	1.0000	22
10	1.0000	0.8929	0.9434	28
11	0.9375	0.6000	0.7317	25
12	0.9200	0.9200	0.9200	25
13	0.9302	1.0000	0.9639	40
14	1.0000	0.9200	0.9583	25
15	0.8400	0.9130	0.8750	23
16	0.9200	0.9583	0.9388	24
17	0.9355	0.9667	0.9508	30
18	0.9615	1.0000	0.9804	25
19	0.9688	0.9688	0.9688	32
20	0.9200	0.8214	0.8679	28
21	0.9167	0.8800	0.8980	25
22	0.9310	0.9643	0.9474	28
23	0.9032	0.9333	0.9180	30
24	0.8800	0.9565	0.9167	23
25	1.0000	1.0000	1.0000	32
26	1.0000	1.0000	1.0000	21
27	1.0000	1.0000	1.0000	25
28	1.0000	0.9000	0.9474	20
29	0.7778	0.8750	0.8235	32
30	0.9231	0.9600	0.9412	25
31	0.9600	1.0000	0.9796	24
32	1.0000	0.9688	0.9841	32
33	0.9524	0.9524	0.9524	21
34	1.0000	0.9333	0.9655	30
35	0.8077	0.8077	0.8077	26
36	1.0000	0.8148	0.8980	27
37	0.9286	0.9630	0.9455	27

38	0.7727	0.7083	0.7391	24
39	0.9333	0.9655	0.9492	29
40	0.8261	0.9500	0.8837	20
41	0.8529	0.9355	0.8923	31
42	0.7941	1.0000	0.8852	27
43	0.9524	1.0000	0.9756	20
44	0.9706	0.9706	0.9706	34
45	0.8182	0.6923	0.7500	26
46	0.9500	0.7917	0.8636	24
47	0.9583	1.0000	0.9787	23
48	0.7407	0.8000	0.7692	25
49	0.9565	0.8462	0.8980	26
Accuracy		0.9252		1337
Macro Avg	0.9267	0.9230	0.9226	1337
Weighted Avg	0.9276	0.9252	0.9242	1337

CONCLUSION

The results of this study demonstrate that hybrid architectures combining convolutional feature extractors with Transformer-based temporal modeling provide a clear advantage over recurrent approaches for multi-class video classification. In particular, the superior performance of the ResNet-Transformer model highlights the effectiveness of self-attention mechanisms in capturing long-range temporal dependencies and improving class-wise consistency. Future work can investigate deeper and more expressive backbones, such as ResNet50 or EfficientNet, which may further enhance feature representation, particularly for visually complex classes. Architectures such as Vision Transformer (ViT), TimeSformer, or Swin Transformer offer improved scalability and hierarchical attention mechanisms, which may better capture both local and global spatiotemporal patterns.Additionally, increasing the overall dataset size is likely to further improve generalization performance. Another promising direction is the incorporation of multimodal information, such as combining visual features with audio or textual metadata, which could improve discrimination in classes with subtle visual differences.

REFERENCES

1. CNNs, RNNs and Transformers in Human Action Recognition: A Survey and a Hybrid Model. https://arxiv.org/html/2407.06162v2.
2. M Ananda Kumar, S Senthilkumar, and S Rajalakshmi. New Deep Learning Video-based Human Activity Recognition using DCNN and RNN in an Edge Computing Environment. *Optik*, page 172757, April 2026.
3. Joao Carreira and Andrew Zisserman. Quo Vadis, Action Recognition? A New Model

and the Kinetics Dataset. https://arxiv.org/abs/1705.07750v3, May 2017.

4. Joao Carreira and Andrew Zisserman. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4733, Honolulu, HI, July 2017. IEEE.
5. Sharana Dharshikgan Suresh Dass, Hrishav Bakul Barua, Ganesh Krishnasamy, Raveendran Paramesran, and Raphael C.-W. Phan. ActNetFormer: Transformer-ResNet Hybrid Method for Semi-Supervised Action Recognition in Videos, April 2024.
6. Sehwan Heo, Junbeom Moon, and Soon Ki Jung. Action recognition: A comprehensive survey of tasks, methods, and challenges. *ICT Express*, 12(1):32–49, February 2026.
7. B. Aruna Kumari, M. Bhargavi, B. Aswini, N. Yamini, and K. Vedavathi. Human Action Recognition from Video Frames Using Recurrent Neural Network. In *2024 2nd International Conference on Disruptive Technologies (ICDT)*, pages 731–737, March 2024.
8. Chuankun Li, Pichao Wang, Shuang Wang, Yonghong Hou, and Wanqing Li. Skeleton-based Action Recognition Using LSTM and CNN, July 2017.
9. Pradyumn Patil, Vishwajeet Pawar, Yashraj Pawar, and Shruti Pisal. Video Content Classification using Deep Learning, November 2021.
10. Hieu H Pham, Louahdi Khoudour, Alain Crouzil, Pablo Zegers, and Sergio A Velastin. Video-based Human Action Recognition using Deep Learning: A Review.