technologies

MDPI

Article

A New Efficient Hybrid Technique for Human Action Recognition Using 2D Conv-RBM and LSTM with Optimized Frame Selection

Majid Joudaki [1,*], Mehdi Imani [2,*] and Hamid R. Arabnia [3]

[1] Electrical and Computer Engineering, University of Kashan, Kashan 8731753153, Iran
[2] Department of Computer and System Sciences, Stockholm University, 10691 Stockholm, Sweden
[3] School of Computing, University of Georgia, Athens GA 30602, USA; hra@uga.edu
* Correspondence: m.joudaki@gmail.com (M.J.); m.imani@gmail.com (M.I.)

Abstract: Recognizing human actions through video analysis has gained significant attention in applications like surveillance, sports analytics, and human–computer interaction. While deep learning models such as 3D convolutional neural networks (CNNs) and recurrent neural networks (RNNs) deliver promising results, they often struggle with computational inefficiencies and inadequate spatial–temporal feature extraction, hindering scalability to larger datasets or high-resolution videos. To address these limitations, we propose a novel model combining a two-dimensional convolutional restricted Boltzmann machine (2D Conv-RBM) with a long short-term memory (LSTM) network. The 2D Conv-RBM efficiently extracts spatial features such as edges, textures, and motion patterns while preserving spatial relationships and reducing parameters via weight sharing. These features are subsequently processed by the LSTM to capture temporal dependencies across frames, enabling effective recognition of both short- and long-term action patterns. Additionally, a smart frame selection mechanism minimizes frame redundancy, significantly lowering computational costs without compromising accuracy. Evaluation on the KTH, UCF Sports, and HMDB51 datasets demonstrated superior performance, achieving accuracies of 97.3%, 94.8%, and 81.5%, respectively. Compared to traditional approaches like 2D RBM and 3D CNN, our method offers notable improvements in both accuracy and computational efficiency, presenting a scalable solution for real-time applications in surveillance, video security, and sports analytics.

Keywords: action recognition; convolutional restricted Boltzmann machine; long short-term memory; spatial–temporal feature extraction; video processing

Academic Editors: Pedro Antonio Gutiérrez and Mohammed Mahmoud

Received: 27 November 2024
Revised: 19 January 2025
Accepted: 27 January 2025
Published: 1 February 2025

Citation: Joudaki, M.; Imani, M.; Arabnia, H.R. A New Efficient Hybrid Technique for Human Action Recognition Using 2D Conv-RBM and LSTM with Optimized Frame Selection. *Technologies* **2025**, *13*, 53. https://doi.org/10.3390/technologies13020053

1. Introduction

The field of video-based human action recognition has garnered significant attention due to its wide-ranging applications in domains such as surveillance, sports analytics, human–computer interaction, and healthcare monitoring [1,2]. Recognizing human actions in real-time from video data is challenging because of the high dimensionality of video frames, complex motion patterns, and the need for effective spatial–temporal data understanding [3]. Traditional approaches using handcrafted features often fail to capture the intricate spatial–temporal relationships inherent in human actions [4].

Recent advancements in deep learning have revolutionized video analysis by enabling automated feature extraction directly from raw data. However, these methods face challenges in computational efficiency and scalability, especially for high-resolution

or long-duration video sequences [5,6]. As noted in [7], deep learning's pivotal role in machine and robotic vision has driven significant progress in areas such as object detection, semantic segmentation, and action recognition. This underscores the necessity for models capable of robustly handling the spatial–temporal complexities of video data.

Among the most successful approaches are hybrid models leveraging convolutional neural networks (CNNs) and recurrent neural networks (RNNs), particularly long short-term memory (LSTM) networks [8]. CNNs are effective in extracting spatial features from individual video frames, learning local patterns such as edges and textures [9]. Meanwhile, LSTMs excel at capturing temporal dependencies, retaining information about past frames to recognize sequential patterns [1].

Despite their promise, CNN-LSTM architectures often encounter challenges with computational inefficiency due to the high number of parameters and resource demands. These challenges become particularly pronounced with high-resolution data or extended video sequences [2]. Additionally, CNNs, while powerful for static image analysis, may not fully capture the dynamic nature of motion over time, limiting their effectiveness in spatial–temporal feature extraction [10]. This has driven the exploration of alternative architectures that balance accuracy and computational efficiency [11].

Recently, vision transformers (ViTs) have emerged as a promising alternative for action recognition tasks. Unlike CNNs, which rely on local receptive fields, ViTs utilize self-attention mechanisms to model global dependencies across spatial and temporal dimensions. This enables them to capture complex relationships between features that span across the entire video frame [12]. ViTs have demonstrated state-of-the-art performance in several visual tasks due to their ability to process sequences of image patches as tokens, treating each patch as an individual input unit [13]. For video action recognition, models such as ViViT (video vision transformer) [14] and TimeSformer [15] have extended the transformer framework to temporal data, effectively learning spatial–temporal representations. However, ViT-based models often require significant computational resources and large-scale pretraining on video datasets, which can limit their scalability and accessibility [16].

Restricted Boltzmann machines (RBMs), particularly two-dimensional RBMs (2D RBMs), have recently been revisited for their ability to learn complex distributions and hierarchical features in an unsupervised manner [4]. Unlike traditional RBMs, 2D RBMs can better preserve local pixel relationships, making them suitable for spatial data such as video frames. However, their inability to model temporal dependencies across frames limits their application in action recognition tasks where motion dynamics are essential [4].

To address these limitations, this paper proposes a novel hybrid architecture combining two-dimensional convolutional RBMs (2D Conv-RBMs) and LSTM networks. The 2D Conv-RBM incorporates convolutional filters into the RBM framework, enabling efficient extraction of spatial features such as edges, textures, and motion cues while reducing parameters through weight sharing. These spatial features are then processed by an LSTM layer, which captures temporal dependencies across frames, enabling robust recognition of both short-term and long-term action patterns.

A notable aspect of this work is the adoption of a smart frame selection mechanism, originally introduced in prior research, which has been effectively integrated into our proposed method. This mechanism reduces redundancy by selecting only the most informative frames for processing, significantly lowering computational costs without sacrificing model accuracy. By focusing on key temporal transitions, this method enhances the network's ability to capture critical dynamics in video sequences.

The primary contributions of this paper are as follows:

1. We introduce a novel hybrid 2D Conv-RBM + LSTM architecture that efficiently captures both spatial and temporal features for action recognition tasks. By leveraging the strengths of unsupervised spatial feature learning through Conv-RBM and temporal modeling through LSTM, the proposed method achieves robust and effective action recognition.
2. We incorporate a smart frame selection mechanism that reduces computational complexity by selecting only the most relevant frames in each video sequence. This innovation minimizes redundancy while preserving critical temporal information, enabling the network to focus on the most informative portions of the video.
3. We conduct extensive evaluations on three benchmark datasets: KTH [17], UCF Sports [18], and HMDB51 [19]. On the KTH and UCF Sports datasets, our method achieves state-of-the-art accuracy, surpassing all competing methods in the literature. On the HMDB51 dataset, while our method achieves competitive accuracy, certain other approaches demonstrate higher performance, particularly those leveraging transformer-based architectures or highly complex deep learning frameworks. Despite this, our method balances accuracy and computational efficiency, making it a promising solution for real-time action recognition tasks.

The remainder of this paper is organized as follows: Section 2 reviews related work in video-based action recognition and spatial–temporal feature extraction. Section 3 presents the detailed architecture of the proposed model and the smart frame selection mechanism. Section 4 describes the experimental setup, including datasets and metrics. Section 5 discusses the results and analysis, and Sections 6 and 7 concludes the paper with potential future directions.

2. Related Work

Human action recognition from video sequences has long been a challenging problem in the field of computer vision. Early methods relied on handcrafted features such as histogram of oriented gradients (HOG) and optical flow to extract motion and appearance cues from videos. While effective in some cases, these traditional techniques often struggle to capture the complex spatial–temporal dynamics present in human actions. With the rise of deep learning, convolutional neural networks (CNNs) and recurrent neural networks (RNNs), especially long short-term memory (LSTM) networks, have dominated the field, offering more robust and automatic feature extraction and sequence modeling capabilities [1,6]. Additionally, the integration of mobile and embedded sensors, as demonstrated by [20] in their smartphone-based motion detection model, has opened new avenues for real-time and mobile applications of human activity recognition, further highlighting the adaptability of deep learning in diverse environments.

2.1. CNN-Based Approaches for Spatial Feature Extraction

CNNs have been widely adopted in human action recognition due to their powerful capability in extracting spatial features from video frames. The foundational work by [21] introduced the two-stream CNN model, which processes both spatial (static frame) and temporal (optical flow) streams to recognize actions, emphasizing the importance of combining spatial and temporal information for video analysis [22]. Recent advancements have expanded on CNN-based approaches, with models like inflated 3D ConvNet [23] inflating 2D CNNs into 3D convolutions to capture spatial–temporal features simultaneously across video frames [24]. While these models exhibit strong performance, they come with increased computational complexity due to the higher number of parameters associated with 3D convolutions, which can limit real-time applicability [25]. The proposed method by [26] fuses spatial and temporal features learned from a principal component analysis

network (PCANet) with bag-of-features (BoF) and vector of locally aggregated descriptors (VLAD) encoding schemes for human action recognition. The method described in [27] is a spatial–temporal interaction learning two-stream (STILT) network for action recognition, which integrates an alternating co-attention mechanism within a two-stream structure (spatial and temporal streams) to optimize spatial and temporal feature interactions, enabling improved recognition accuracy by leveraging complementary information from RGB frames and optical flow.

2.2. LSTM Networks for Temporal Dependencies

Although CNNs are effective for spatial feature extraction, they have inherent limitations in modeling temporal dependencies across video frames. LSTM networks, designed to capture long-term dependencies, address these limitations through their internal memory units. Ref. [28] introduced the LRCN (long-term recurrent convolutional networks) model, combining CNNs for feature extraction with LSTMs for sequence modeling. This approach demonstrated the power of LSTMs in learning temporal dependencies across sequences, and since then, CNN-LSTM combinations have become a standard in video action recognition tasks [1]. More recently, ref. [29] proposed an attention-enhanced CNN-LSTM model that focuses on both key spatial features and significant temporal segments within a video. This use of attention mechanisms helps to filter out irrelevant information, which aligns with the smart frame selection concept utilized in our proposed model [30].

2.3. Restricted Boltzmann Machines (RBMs) and Conv-RBM Variants

Restricted Boltzmann machines (RBMs) have seen varied applications in deep learning, especially for unsupervised feature learning. While traditional RBMs were originally used to capture dependencies within static images by learning latent representations from raw pixel data, they are limited by their fully connected nature, which hinders spatial coherence and computational efficiency for large-scale image and video data [4]. To overcome these challenges, two-dimensional RBMs (2D RBMs) were introduced, preserving local pixel relationships in video frames to maintain spatial coherence [4]. However, standard 2D RBMs still suffer from inefficiencies due to the lack of parameter sharing. Convolutional RBMs (Conv-RBMs) improve upon this by applying convolutional filters within the RBM framework, generating multiple feature maps, and capturing various spatial patterns with fewer parameters through weight sharing.

Conv-RBMs thus present an efficient method for tasks like action recognition, where spatial structure is critical, as they generate localized feature maps that efficiently handle large-scale data [31,32]. While Conv-RBMs are relatively new in video-based action recognition, our proposed architecture combines Conv-RBMs with LSTM networks to enhance both spatial and temporal dependencies. This combination aligns well with recent advancements in skeleton-based activity recognition, such as [33], who used autoencoders for feature extraction, further reinforcing the potential of unsupervised learning models in human action recognition.

2.4. Comparison Between Conv-RBM and CNN

Convolutional restricted Boltzmann machines (Conv-RBMs) and convolutional neural networks (CNNs) are widely utilized for spatial feature extraction in image and video analysis. Despite their shared reliance on convolutional operations, the two approaches differ significantly in their architecture, learning paradigms, and applications.

Conv-RBMs, a variant of restricted Boltzmann machines (RBMs), are generative energy-based models designed to learn hierarchical representations in an unsupervised manner [34]. They model the joint probability distribution of visible and hidden units

using an energy function, as described in Equation (1) in Section 3.2. By incorporating convolutional filters into their structure, Conv-RBMs enable efficient extraction of localized spatial features while preserving critical relationships between neighboring pixels. These models employ weight sharing across receptive fields, significantly reducing the number of parameters compared to traditional RBMs or fully connected networks [35]. As described in Equation (4), Section 3.2, the probabilistic activation of hidden units depends on the convolutional interaction between the input and the learned filters. The unsupervised nature of Conv-RBMs makes them particularly advantageous for tasks where labeled data is scarce or expensive to obtain, as they can effectively learn meaningful features directly from raw data.

In contrast, CNNs are discriminative, supervised models that excel in classification tasks by optimizing parameters through backpropagation based on labeled data [36]. CNNs use convolutional layers to extract spatial hierarchies of features, such as edges and textures, followed by pooling layers to reduce spatial dimensions. While highly effective in feature extraction, CNNs require substantial labeled data and computational resources to achieve optimal performance. Furthermore, CNNs are inherently limited by their focus on learning task-specific features, making them less flexible for unsupervised or semi-supervised learning scenarios.

One of the key differences lies in their learning mechanisms. Conv-RBMs optimize an energy function to learn latent representations, enabling them to capture generalizable and compact features. This generative approach contrasts with the purely discriminative nature of CNNs, which focus on minimizing classification error. As a result, Conv-RBMs tend to produce more interpretable and transferable feature representations [37]. Additionally, Conv-RBMs are better suited for capturing localized pixel dependencies, which are crucial for understanding motion patterns and spatial relationships in video frames. This capability is especially beneficial for human action recognition tasks, where subtle variations in motion and appearance play a critical role.

From a computational perspective, Conv-RBMs are lightweight due to their parameter-sharing mechanism, making them more suitable for scenarios with limited resources. In contrast, CNNs typically require higher computational power, especially when working with high-resolution images or large-scale datasets. However, CNNs benefit from a mature ecosystem of pre-trained models and frameworks, which can be fine-tuned for specific applications.

In the context of this work, Conv-RBM was chosen over CNN for spatial feature extraction due to its ability to operate in an unsupervised manner while preserving local spatial coherence. This property is critical for human action recognition, where spatial features need to be generalized across diverse video frames before temporal dependencies can be modeled. Moreover, Conv-RBM's efficient parameterization aligns well with the smart frame selection mechanism employed in the proposed method, further enhancing computational efficiency without sacrificing accuracy.

2.5. Smart Frame Selection in Video Analysis

One of the major challenges in video-based action recognition is the large number of frames in video sequences, many of which are redundant or uninformative. Processing every frame is computationally costly, especially for real-time applications. Smart frame selection techniques address this by identifying and selecting only the most informative frames, reducing computational cost without compromising accuracy [38]. Ref. [20] demonstrated the impact of frame selection in mobile action recognition, where computational efficiency is crucial due to hardware constraints.

Several methods have been proposed for smart frame selection. Dynamic selection techniques have been employed to optimize key frame selection based on motion clustering, enabling efficient video abstraction and representation [39]. Techniques such as clustering wavelet coefficients and using Jensen–Shannon divergence have proven effective in segmenting video content and extracting representative key frames [40,41]. Our proposed model extends the smart frame selection approach presented in [42] with a Conv-RBM + LSTM architecture, ensuring that the network focuses on the most relevant temporal information and reducing the computational overhead, making it suitable for real-time applications [42].

2.6. Benchmark Datasets and Evaluation

Performance in action recognition is often evaluated on benchmark datasets such as KTH, UCF Sports, and HMDB51. These datasets provide a diverse range of human activities, from simple actions (e.g., walking and clapping in KTH) to complex sports activities (e.g., in UCF Sports) and varied real-world actions (HMDB51). Ref. [43] demonstrated high accuracy on these datasets using 3D CNNs combined with attention mechanisms, highlighting the strength of deep learning approaches for complex video analysis [44]. However, the high computational cost of these methods underscores the need for more efficient architectures, like the one proposed in this paper.

3. Proposed Method

This section presents a novel architecture for video-based human action recognition, integrating smart frame selection, two-dimensional convolutional restricted Boltzmann machine (2D Conv-RBM) for spatial feature extraction, and long short-term memory (LSTM) for temporal modeling. The final features are classified through a fully connected network. This pipeline addresses the challenges of redundant video frames, ensuring efficient computational processing and enhanced accuracy for real-time action recognition.

The input to the network pipeline is a sequence of video frames with dimensions [s, c, h, w], where s represents the number of frames, c is the number of channels (converted to grayscale, c = 1), and $h \times w$ denotes the spatial resolution of each frame. First, the sequence undergoes preprocessing, where each frame is resized to 64×64 for uniformity and computational efficiency. Following this, the smart frame selection mechanism identifies the top K frames based on their discriminative importance, reducing the sequence length from s to K. These selected frames, now of dimensions [K, 1, h′, w′], are passed into the 2D Conv-RBM layer, where convolutional filters extract spatial features, producing f feature maps for each frame. After max pooling is applied to reduce spatial dimensions, the output feature maps are transformed into [K, f, h′, w′], where $h' \times w'$ are the reduced dimensions after pooling. The feature maps are then flattened into a compact representation of size [K, f], where $f = n \times h' \times w'$. This sequence is processed by the LSTM layer, which captures temporal dependencies, generating a hidden state of size [U], where U is the number of LSTM units. Finally, the hidden state is fed into a fully connected layer with a softmax activation function, producing a probability distribution over the action classes and outputting the final classification result of size [C]. This pipeline ensures efficient spatial and temporal feature extraction while maintaining computational efficiency. The described pipeline is illustrated in Figure 1.

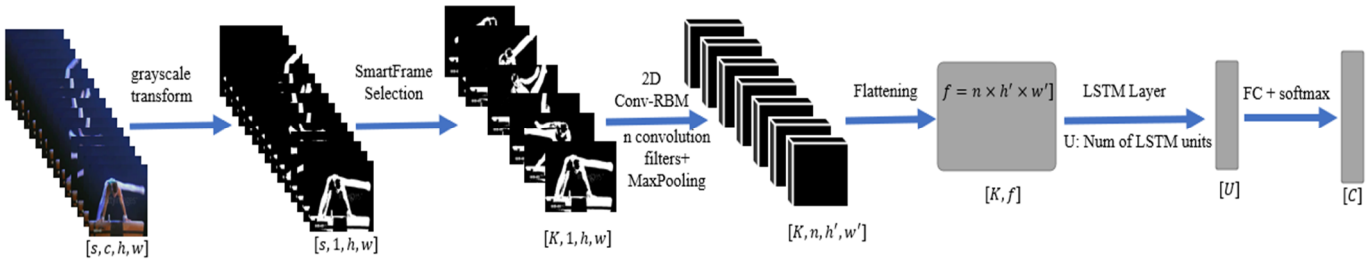


Figure 1. Overview of the proposed action recognition with respect to data dimension changes throughout the network. The pipeline of the proposed method, including preprocessed video frames, smart frame selection, 2D Conv-RBM for spatial feature extraction, LSTM for temporal modeling, and a fully connected layer for action classification.

3.1. Preprocessing and Frame Selection

Before processing, each video sequence is converted into grayscale frames to reduce computational complexity while maintaining the essential features required for action recognition. Grayscale conversion simplifies the input data by reducing dimensionality without compromising critical information related to motion and spatial structure. Once the video frames are preprocessed into grayscale, we apply a smart frame selection mechanism to eliminate redundancy and retain only the most informative frames for further analysis. This mechanism significantly reduces computational costs and ensures that the network processes only the frames representing key temporal transitions, thereby enhancing the efficiency of the action recognition pipeline.

The smart frame selection mechanism, inspired by the method proposed in [38], assesses both the individual and relational importance of video frames. This method consists of two key components: a single-frame selector and a global selector. The single-frame selector examines the information of each frame independently and assigns a score δ_i to indicate the usefulness of the frame for classification. Concurrently, the global selector considers the entire video sequence to capture relationships between frames using an attention and relation network. This network takes pairs of frames as input, represented as concatenated feature vectors, and outputs scores γ_i, reflecting the importance of the temporal relationship between these frames.

The relational network utilizes an attention mechanism to capture temporal changes within actions, considering how frames contribute to the overall action representation. To achieve this, the input sequence $X = (X_1, \ldots, X_N)$, where X_i represents the feature vector of frame i, is augmented by randomly pairing each frame X_i with another frame X_{ri}, sampled from subsequent frames within the sequence. This ensures flexibility in capturing temporal variations, as some actions are better represented by frames that are closely spaced, while others benefit from greater temporal distances. The concatenated vectors $\mathbf{Z}_i = [X_i: X_{ri}]$ are fed into the relational model, which produces temporal relation–attention weights γ_1, γ_2, $\ldots$, γ_N, providing a global representation of the video's temporal structure.

The final discriminative score for each frame is computed by multiplying δ_i and γ_i, resulting in a "goodness" score for each frame. Based on these scores, the top n frames with the highest scores are selected and passed to the spatial–temporal modeling network for classification. This selective approach ensures that only the most critical frames are used for action recognition, reducing computational demands while preserving classification accuracy.

Figure 2 provides an overview of this smart frame selection process, demonstrating how the combination of frame-level and video-level evaluations identifies the most informative frames. The attention and relational modules, fully detailed in [38], are beyond the scope of this work but remain integral to the success of this preprocessing step. By

leveraging this mechanism, our method achieves efficient frame selection without compromising the quality of the spatial–temporal features provided to the subsequent layers of the network.

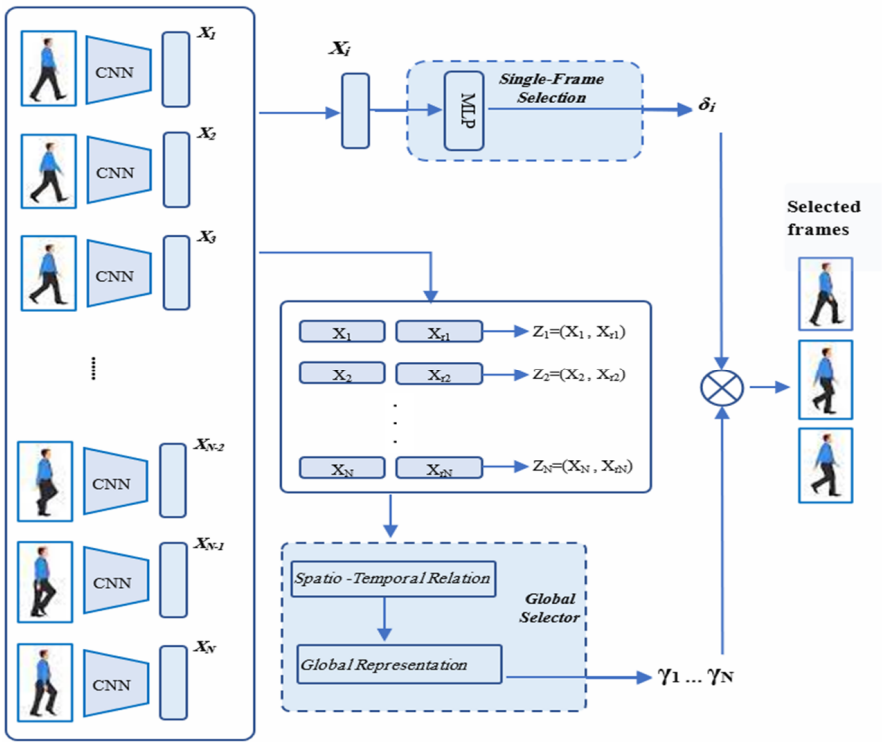


Figure 2. Overview of the smart frame selection mechanism adapted from [38]. This figure illustrates the process of evaluating individual and relational frame importance using both single-frame and global selectors. It showcases the calculation of δ_i and γ_i, combining them to score frame importance and selecting the top n frames based on these scores. The method reduces the number of frames passed to the network while retaining those most critical for action recognition.

3.2. Two-Dimensional Convolutional Restricted Boltzmann Machine (2D Conv-RBM)

The two-dimensional convolutional restricted Boltzmann machine (2D Conv-RBM) is an extension of the traditional restricted Boltzmann machine (RBM), specifically designed to handle spatially structured data like images or video frames. Unlike standard RBMs, which employ fully connected visible and hidden units, the 2D Conv-RBM uses convolutional filters to connect the visible layer $\boldsymbol{V}$ (input frames) to multiple hidden feature maps $\boldsymbol{H}^f$. This architecture preserves local spatial relationships in the input, enabling efficient feature extraction while reducing the number of trainable parameters. The visible layer $\boldsymbol{V}$ represents grayscale frames with dimensions $\boldsymbol{H} \times \boldsymbol{W}$, while the hidden layer comprises multiple feature maps, each learning distinct spatial features. Convolutional filters $\boldsymbol{W}^f$ are shared across spatial locations, ensuring spatial invariance in the learned features.

The relationship between the visible and hidden layers is defined through an energy function $\boldsymbol{E(V, H)}$, which measures the compatibility between the two layers. The joint probability distribution of the visible and hidden units is given by Equation (1):

$$P(\boldsymbol{V},\boldsymbol{H}) = \frac{1}{\boldsymbol{Z}} e^{-E(\boldsymbol{V},\boldsymbol{H})}, \tag{1}$$

where $\boldsymbol{Z}$ is the partition function, summing over all possible configurations of $\boldsymbol{V}$ and $\boldsymbol{H}$:

$$Z = \sum_{\boldsymbol{V},\boldsymbol{H}} e^{-E(\boldsymbol{V},\boldsymbol{H})}, \tag{2}$$

The energy function ***E(V, H)***, shown in Equation (3), is expressed as

$$E(V,H) = -\sum_{f}\sum_{i,j}\left(\sum_{k,l} \boldsymbol{W}_{k,l}^{f} \boldsymbol{V}_{i+k,j+l}\right)\boldsymbol{H}_{i,j}^{f} - \sum_{i,j} \boldsymbol{b}_{i,j}\boldsymbol{V}_{i,j} - \sum_{f}\sum_{i,j} \boldsymbol{c}^{f}\boldsymbol{H}_{i,j}^{f}, \tag{3}$$

where $\boldsymbol{W}_{k,l}^{f}$ represents the convolutional filter connecting the visible and hidden layers, $\boldsymbol{b}_{i,j}$ is the bias for visible units, and $\boldsymbol{c}^{f}$ is the bias for the hidden feature maps. The activation of hidden units is governed by the conditional probability of a hidden unit $\boldsymbol{H}_{i,j}^{f}$ being active (set to 1) given the visible layer $\boldsymbol{V}$, as shown in Equation (4):

$$P(\boldsymbol{H}_{i,j}^{f} = 1|\boldsymbol{V}) = \sigma\left(\sum_{k,l} \boldsymbol{W}_{k,l}^{f}\boldsymbol{V}_{i+k,j+l} + \boldsymbol{c}^{f}\right), \tag{4}$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$ is the sigmoid activation function. Similarly, the visible layer can be reconstructed from the hidden units using the conditional probability in Equation (5):

$$P\left(\boldsymbol{V}_{i,j} = 1\middle|\boldsymbol{H}\right) = \sigma\left(\sum_{f}\sum_{k,l} \boldsymbol{W}_{k,l}^{f}\boldsymbol{H}_{i-k,j-l}^{f} + \boldsymbol{b}_{i,j}\right). \tag{5}$$

The feature map $\boldsymbol{F_{i,j}}$ extracted at location (i, j) is computed as Equation (6):

$$\boldsymbol{F}_{i,j} = \sigma\left(\sum_{k,l} \boldsymbol{W}_{k,l}\boldsymbol{V}_{i+k,j+l} + b\right), \tag{6}$$

where $\boldsymbol{W_{k,l}}$ is the convolutional filter, $\boldsymbol{V}$ is the input frame, and b is the bias term.

The model learns its parameters (weights and biases) using contrastive divergence, an efficient gradient-based learning approach. The weight updates are computed as shown in Equation (7):

$$\frac{\partial \log P(\boldsymbol{V})}{\partial \boldsymbol{W}_{k,l}^{f}} = \left\langle \boldsymbol{V}_{i,j}\boldsymbol{H}_{i+k,j+l}^{f}\right\rangle_{\text{data}} - \left\langle \boldsymbol{V}_{i,j}\boldsymbol{H}_{i+k,j+l}^{f}\right\rangle_{\text{model}} \tag{7}$$

where $\langle\cdot\rangle_{\text{data}}$ and $\langle\cdot\rangle_{\text{model}}$ represent expectations under the data distribution and the model distribution, respectively. Similarly, the updates for visible and hidden biases are computed using Equations (8) and (9):

$$\frac{\partial \log P(\boldsymbol{V})}{\partial \boldsymbol{b}_{i,j}} = \left\langle \boldsymbol{V}_{i,j}\right\rangle_{\text{data}} - \left\langle \boldsymbol{V}_{i,j}\right\rangle_{\text{model}} \tag{8}$$

$$\frac{\partial \log P(\boldsymbol{V})}{\partial \boldsymbol{c}^{f}} = \left\langle \boldsymbol{H}_{i,j}^{f}\right\rangle_{\text{data}} - \left\langle \boldsymbol{H}_{i,j}^{f}\right\rangle_{\text{model}}. \tag{9}$$

To further enhance efficiency, the feature maps generated by the Conv-RBM layer are processed using a pooling layer, such as max-pooling, to downsample the spatial dimensions. This step reduces computational complexity while preserving the most salient features, ensuring that critical information is retained for downstream tasks. Overall, the 2D Conv-RBM effectively captures localized spatial features such as edges and textures, making it highly suitable for action recognition tasks where preserving spatial coherence is essential. This approach is informed by foundational work on energy-based models and convolutional adaptations of RBMs, including studies by [34,35,45,46].

3.3. Long Short-Term Memory (LSTM) for Temporal Modeling

The long short-term memory (LSTM) network plays a critical role in modeling temporal dependencies across video frames after the spatial features have been extracted by the

2D Conv-RBM. LSTMs are particularly well suited for handling sequential data, such as video frames, due to their ability to capture both short-term and long-term dependencies. This capability is achieved through an internal gating mechanism that controls the flow of information, allowing the network to selectively remember or forget information at each time step. The LSTM maintains a memory cell state C_t, which is updated iteratively as it processes each frame, enabling the modeling of complex temporal patterns in human actions [36].

At each time step t, the LSTM receives an input vector $\boldsymbol{x}_t$, which in this case is the spatial feature vector generated by the 2D Conv-RBM. The hidden state from the previous time step $\boldsymbol{h}_{t-1}$ is combined with $\boldsymbol{x}_t$ to compute the values of the gates and update the cell state. The first gating mechanism, the forget gate, determines how much of the previous cell state $\boldsymbol{C}_{t-1}$ should be retained. The forget gate is computed as Equation (10):

$$\boldsymbol{f}_t = \sigma\Big(\boldsymbol{W}_f[\boldsymbol{h}_{t-1}, \boldsymbol{x}_t] + \boldsymbol{b}_f\Big) \tag{10}$$

where $\boldsymbol{W}_f$ and $\boldsymbol{b}_f$ are the weights and biases associated with the forget gate, and $\sigma(x)$ is the sigmoid activation function.

Next, the input gate decides how much new information should be written to the memory cell. The input gate is computed as

$$\boldsymbol{i}_t = \sigma(\boldsymbol{W}_i[\boldsymbol{h}_{t-1}, \boldsymbol{x}_t] + \boldsymbol{b}_i) \tag{11}$$

and the candidate cell state $\widetilde{\boldsymbol{C}}_t$, which represents new information to be added, is calculated as

$$\widetilde{\boldsymbol{C}}_t = \tanh(\boldsymbol{W}_C[\boldsymbol{h}_{t-1}, \boldsymbol{x}_t] + \boldsymbol{b}_C) \tag{12}$$

The cell state $\boldsymbol{C}_t$ is then updated by combining the retained information from the previous cell state (modulated by the forget gate) with the newly computed candidate cell state (modulated by the input gate):

$$\boldsymbol{C}_t = \boldsymbol{f}_t * \boldsymbol{C}_{t-1} + \boldsymbol{i}_t * \widetilde{\boldsymbol{C}}_t \tag{13}$$

The output gate determines the information to be propagated to the hidden state $\boldsymbol{h}_t$, which is used for the next time step or for making predictions. The output gate is calculated as

$$\boldsymbol{o}_t = \sigma(\boldsymbol{W}_o[\boldsymbol{h}_{t-1}, \boldsymbol{x}_t] + \boldsymbol{b}_o) \tag{14}$$

and the hidden state is then updated using the current cell state and the output gate:

$$\boldsymbol{h}_t = \boldsymbol{o}_t * \tanh(\boldsymbol{C}_t) \tag{15}$$

In these equations, $\boldsymbol{W}_i$, $\boldsymbol{W}_C$, $\boldsymbol{W}_o$ and $\boldsymbol{b}_i$, $\boldsymbol{b}_C$, $\boldsymbol{b}_o$ are the weights and biases associated with the input, candidate, and output gates, respectively.

The LSTM enables the network to retain important information over long sequences of frames while discarding irrelevant details, ensuring that both short-term and long-term dependencies are effectively captured. This property makes LSTMs particularly well suited for action recognition tasks, where sequential patterns in video frames are critical for accurately classifying human actions. By combining the spatial feature extraction of the 2D Conv-RBM with the temporal modeling of the LSTM, the proposed architecture achieves robust performance on complex video-based tasks.

3.4. Feature Classification Using Fully Connected Network

The final step in the proposed pipeline involves the classification of features extracted by the LSTM. At the last time step of the LSTM, the final hidden state h is obtained, which encodes both spatial and temporal information relevant to the action sequence. This feature vector h is then passed to a fully connected layer, where it is linearly transformed using a weight matrix $\boldsymbol{W}_c$ and a bias term $\boldsymbol{b}_c$. The result of this linear transformation is then fed into a softmax activation function, which outputs a probability distribution over the possible action classes. The classification process is mathematically defined as follows:

$$\boldsymbol{y} = \text{softmax}(\boldsymbol{W}_c\boldsymbol{h} + \boldsymbol{b}_c), \tag{16}$$

where $\boldsymbol{y}$ represents the predicted probability distribution over all action classes. The network is trained to minimize the cross-entropy loss, which measures the difference between the predicted probability distribution and the true labels. The cross-entropy loss function is given by

$$L = -\sum_{i=1}^{N} \boldsymbol{y}_i \log(\hat{\boldsymbol{y}}_i) \tag{17}$$

where $\boldsymbol{y}_i$ is the true label for the *i-th* video, $\hat{\boldsymbol{y}}_i$ is the predicted probability for that class, and N is the total number of training samples. This loss function ensures that the predicted probabilities closely align with the ground truth labels.

To optimize the network, the Adam optimizer is employed due to its adaptive learning rate and efficient convergence properties. Additionally, regularization techniques such as dropout are applied to the fully connected layer to reduce overfitting by randomly deactivating a fraction of the neurons during training. These strategies ensure robust performance of the model, even when applied to complex and diverse action recognition datasets. The combination of the fully connected network and the softmax layer provides a powerful and interpretable mechanism for final action classification.

3.5. Proposed Architecture Specifications

The proposed architecture pipeline ensures efficient and accurate video-based human action recognition by addressing both computational challenges and the need for robust spatial–temporal feature extraction. The method is particularly well suited for real-time applications, such as video surveillance and sports analytics, due to its reduced computational overhead and high accuracy. Based on recent studies and relevant literature, we have gathered and organized the parameters for our proposed model in Table 1. This table includes the details for frame dimensions, visible and hidden layers, filter sizes, LSTM configurations, and learning algorithm parameters.

Table 1. Specifications of the proposed architecture, detailing the parameter configurations for each stage of the pipeline, including 2D Conv-RBM and LSTM layers, optimization settings, and training configurations.

Parameter	Value/Description	Reference
K (smart frame selection)	32 frames	based on [38]
Frame dimensions	64 × 64 (grayscale, black-and-white)	Common practice in video action recognition models
Visible layer size (Conv-RBM)	64 × 64 (corresponding to the frame dimensions)	Based on RBM architecture for spatial extraction
Hidden layer size (Conv-RBM)	64 × 32 × 32 (after pooling)	Reduced spatial dimensions with 64 feature maps

Table 1. *Cont.*

Parameter	Value/Description	Reference
Convolutional filter size (Conv-RBM)	3 × 3 (with stride 1)	Standard in CNNs, balances spatial locality and depth
Pooling layer (Conv-RBM)	2 × 2 (max pooling)	Reduces feature map dimensions by half
LSTM units	256 units	Suitable for temporal modeling of moderate complexity
Number of LSTM layers	2 layers	Allows capturing both short- and long-term dependencies
Optimizer	Adam optimizer (learning rate: 0.001)	Adaptive learning rate method for efficient convergence
Loss function	Cross-entropy loss	Commonly used in classification tasks
Regularization (dropout)	Dropout rate: 0.4 (LSTM layer and FC layer)	Prevents overfitting by dropping 40% of neurons
Learning rate	0.001	Default for Adam, tuned for stability
Batch size	32	Balances between computational load and convergence
Epochs	100	Enough for deep architectures like Conv-RBM + LSTM

4. Experimental Setup

The proposed network was implemented using the PyTorch deep learning framework. The experiments were conducted in a high-performance computing environment featuring two NVIDIA Tesla T4 GPUs, each equipped with 15,360 MiB of memory and running on CUDA version 12.2. The system was optimized for deep learning tasks, ensuring efficient utilization of GPU memory, with both GPUs initialized with 0 MiB memory usage. This setup provided the computational resources required for training and inference on large-scale video datasets in a reasonable timeframe.

The model was evaluated on three widely used human action recognition datasets. The KTH dataset, containing over 600 video sequences across six action classes such as walking and jogging, was captured under controlled conditions. The UCF Sports dataset, comprising 10 action classes involving complex sports activities, introduced challenges related to diverse backgrounds and variations. The HMDB51 dataset, with 51 action classes collected from real-world scenarios, presented significant variations in action execution, background noise, and video quality. To ensure consistency across datasets, preprocessing steps included converting all frames to grayscale to reduce computational complexity, resizing frames to 64 × 64 pixels for uniform input dimensions, and normalizing pixel values to the range (0, 1). Additionally, a smart frame selection mechanism was applied to retain only the most informative frames, reducing redundancy and computational overhead.

The performance of the model was assessed using two key metrics: accuracy and the confusion matrix. Accuracy, which measures the proportion of correctly classified instances, was computed using the following formula:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100 \tag{18}$$

Confusion matrices were generated for each dataset to provide a detailed breakdown of classification performance, including counts of true positives (TP), false positives (FP), false negatives (FN), and true negatives (TN), offering insights into potential misclassifications.

To demonstrate the effectiveness of the proposed model, its performance was compared against several baseline methods. These included 3D CNNs, known for their ability to capture spatial–temporal features; LSTM-based architectures, effective in modeling temporal dependencies; the I3D model, which extends 2D CNNs into the spatial–temporal domain by inflating convolutions; and traditional 2D RBMs, which focus solely on spatial feature extraction and transformer-based models, such as VideoMAE, which leverage self-attention mechanisms to capture global spatial and temporal dependencies. These comparisons highlighted the advantages of the proposed hybrid architecture in combining efficient spatial feature extraction with Conv-RBM and robust temporal modeling with LSTM.

5. Results and Analysis

This section presents a detailed evaluation of the proposed 2D Conv-RBM + LSTM model across the KTH, UCF Sports, and HMDB51 datasets. The analysis includes performance metrics, comparative evaluations, and insights into model behavior.

5.1. Datasets

This section offers a concise overview of the datasets utilized. More detailed explanations are available in the corresponding articles.

5.1.1. KTH

The KTH dataset consists of six types of human actions, including running, jogging, walking, hand waving, hand clapping, and boxing [17]. A total of 600 video sequences are performed by 25 actors, sampled at 25 frames per second (fps), with an original resolution of 160 $\times$ 120 pixels. These videos are recorded in four different scenarios: S1 (outdoors), S2 (outdoors with variations in image scale), S3 (outdoors with different clothing), and S4 (indoors). To reduce computational complexity, the resolution of all images was downscaled to 64 $\times$ 64 pixels.

5.1.2. UCF Sports

The UCF Sports dataset includes ten sports-related actions: diving, golf swinging, lifting, kicking, riding horses, running, skateboarding, swinging on a bench, swinging on a side apparatus, and walking [18]. This dataset comprises approximately 150 video sequences with an original resolution of 720 $\times$ 480 pixels. The videos are collected from broadcast TV channels and are recorded in unrestricted environments, introducing several intra-class variations such as illumination changes, complex backgrounds, motion blur, occlusion, and diverse scene settings. In our experiments, all frames were resized to 64 $\times$ 64 pixels to standardize input dimensions and reduce computational demands. These preprocessing steps enabled a consistent and efficient analysis of the dataset.

5.1.3. HMDB51

The HMDB51 dataset is a widely used benchmark for human action recognition, featuring 51 distinct action classes such as clapping, kicking, jumping, sword exercise, and golf swing [19]. The dataset consists of over 6700 video clips collected from various online sources, including movies and public video archives, making it highly diverse and challenging. Each video clip captures real-world scenarios with significant variations in background, camera motion, lighting, occlusion, and action execution. The videos have a resolution of 240 $\times$ 320 pixels and are recorded at various frame rates. For consistency and computational efficiency in this study, all video frames were resized to 64 $\times$ 64 pixels, and grayscale conversion was applied. This preprocessing step ensured uniformity across datasets while maintaining the essential features for action recognition. The HMDB51

dataset's complexity provides a rigorous testbed for evaluating the proposed method's ability to generalize and handle real-world variations in human actions.

5.2. Reconstruction Error Analysis

The reconstruction error of the 2D Conv-RBM quantifies the model's ability to learn meaningful spatial features by measuring the difference between the input data and the reconstructed visible probabilities. The analytical expression for the reconstruction error is given by

$$E_{reconstruction} = \frac{1}{N}\sum_{i}^{N}\left\|\boldsymbol{X}_i - \hat{\boldsymbol{X}}_i\right\|^2 \tag{19}$$

where

- N is the total number of frames in the batch,
- $\boldsymbol{X}_i$ represents the input data (original frame),
- $\hat{\boldsymbol{X}}_i$ represents the reconstructed visible probabilities of the frame,
- $\|.\|^2$ denotes the squared L2-norm.

This metric provides a direct assessment of the Conv-RBM's capacity to learn spatial representations, with lower reconstruction errors indicating better feature extraction and reconstruction capabilities. As shown in Figure 3, the reconstruction error trends over the course of training reflect the model's adaptability to different datasets. For the KTH dataset, characterized by relatively simple and controlled actions, the reconstruction error started at approximately 12 and rapidly decreased to around 3 within 100 epochs. This swift reduction highlights the model's efficiency in learning spatial features from less complex data. On the other hand, the UCF Sports dataset, involving more dynamic and varied action classes, began with a reconstruction error of approximately 14, which gradually declined to around 6. This indicates the model's steady progress in adapting to more complex motion patterns.

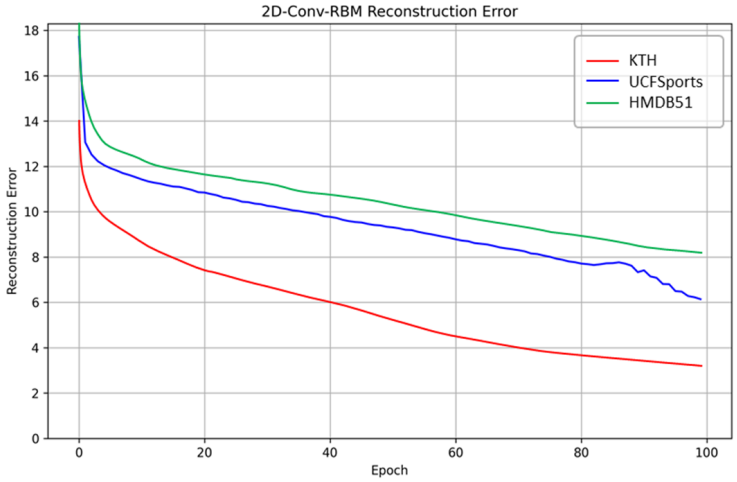


Figure 3. 2D Conv-RBM reconstruction error for all datasets.

The HMDB51 dataset, presenting the greatest challenge due to its diversity and noise, started with an initial reconstruction error of approximately 18. Over the course of training, this error gradually decreased to around 8, reflecting the dataset's complexity and variability. Despite the slower convergence, the steady reduction in reconstruction error demonstrates the robustness of the Conv-RBM in learning spatial representations, even in real-world scenarios with high variability. These trends collectively highlight the model's ability to effectively extract and reconstruct spatial features across datasets of

varying complexity, underscoring the adaptability and generalization capabilities of the 2D Conv-RBM.

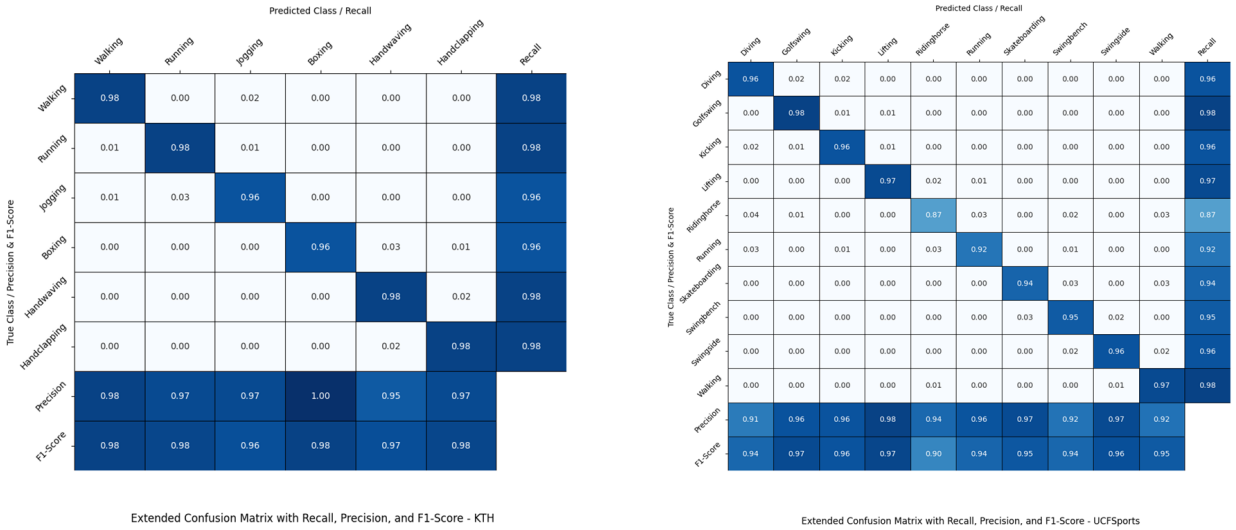
The reconstruction error of the 2D Conv-RBM, as depicted in Figure 3, reflects the model's ability to learn meaningful spatial features over the course of training. For the KTH dataset, which consists of relatively simple and controlled actions, the reconstruction error started at approximately 12 and rapidly decreased to around 3 within 100 epochs. This significant and swift reduction demonstrates the model's efficiency in extracting spatial features from well-structured and less complex data. In contrast, the UCF Sports dataset, which involves more dynamic and varied action classes, exhibited a slower but consistent decline in reconstruction error, starting at approximately 14 and converging at around 6 after 100 epochs. This indicates the model's adaptability to more complex motion patterns while maintaining a steady learning trajectory. The HMDB51 dataset, characterized by its real-world diversity and noise, presented the most challenging case, with an initial reconstruction error of approximately 18 that gradually decreased to around 8 by the end of training. The higher starting error and slower convergence underscore the complexity of this dataset and the variability in its spatial features. Despite these challenges, the Conv-RBM demonstrated its robustness, achieving a steady reduction in error and successfully capturing relevant spatial features across all datasets. These results highlight the model's capability to efficiently learn spatial representations, even in challenging environments with diverse scenarios.

5.3. Performance Metrics: Confusion Matrices and Class-Wise Evaluation

The performance metrics of the proposed model were evaluated using confusion matrices for the KTH, UCF Sports, and HMDB51 datasets, providing a detailed class-wise breakdown of true positives, false positives, and misclassifications. The confusion matrices, depicted for each dataset, highlight the model's accuracy, precision, recall, and F1 scores across all action classes. For the KTH dataset, the model achieved an accuracy of 97.3%, demonstrating near-perfect classification for most action classes. Classes such as walking and boxing achieved F1 scores of 0.98, while slight confusion between running and jogging led to minor dips in performance for these similar motion patterns.

On the UCF Sports dataset, the model attained an accuracy of 94.8%, with strong classification performance across most action categories. However, some misclassifications were observed between visually similar actions, such as riding horse and running, resulting in a lower F1 score of 0.9 for certain classes. This is indicative of the dataset's increased complexity and diversity in background and motion.

For the HMDB51 dataset, which features real-world scenarios with a wide range of variability, the model achieved an accuracy of 81.5%. As the diagonal values (i.e., true positives) are significantly larger than the off-diagonal values, a logarithmic normalization (LogNorm) was applied to the color scale. This adjustment allowed the color differences between the small off-diagonal values and large diagonal values to be visible, providing a better contrast. While simpler actions such as clapping and jumping exhibited strong performance with high F1 scores, complex actions such as sword exercise and golf swing demonstrated some confusion due to overlapping motion patterns, yielding lower F1 scores around 0.78. These results underscore the model's ability to generalize effectively across datasets of varying complexity while highlighting areas for potential improvement in handling fine-grained distinctions among similar actions. The confusion matrices provide crucial insights into the strengths and limitations of the model, aiding in the interpretation of its performance on diverse human action recognition tasks, as depicted in Figures 4 and 5.



(**a**) (**b**)

Figure 4. Extended confusion matrix for KTH (**a**) and UCF Sports (**b**) datasets. This visualization provides a comprehensive assessment of classifier performance, including class-level recall (right column), precision, and F1 score (bottom rows). Darker diagonal elements indicate strong correct predictions, while off-diagonal color contrast reveals false positives and false negatives.

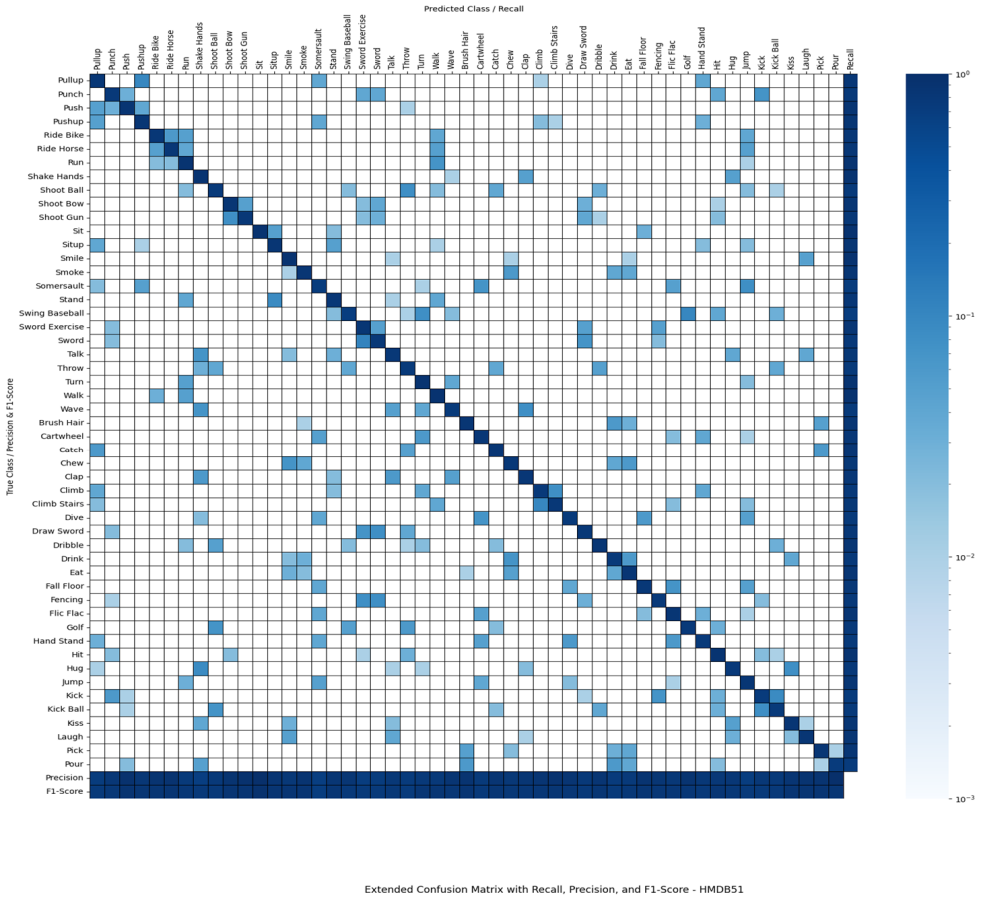


Figure 5. Extended confusion matrix for the HMDB51 dataset. Logarithmic scaling enhances visibility across all value ranges, offering insights into both overall accuracy and class-specific performance.

5.4. LSTM Training Performance

The training performance of the LSTM in capturing temporal dependencies is evident from the trends in training loss and accuracy, as depicted in Figure 6. The red curve represents the training loss, which exhibited a rapid decline during the initial 30 epochs, demonstrating the model's ability to learn temporal patterns efficiently. The loss continued to decrease gradually before converging to minimal values by epoch 100, indicating effective optimization and temporal modeling. In parallel, the green curve reflects the training accuracy, which showed a steady and consistent increase throughout the training process. By epoch 100, the accuracy reached near-maximal values, further confirming the model's robustness in learning complex temporal relationships across all datasets. These trends collectively validate the efficacy of the LSTM in modeling sequential dependencies and ensuring reliable performance in the proposed architecture.

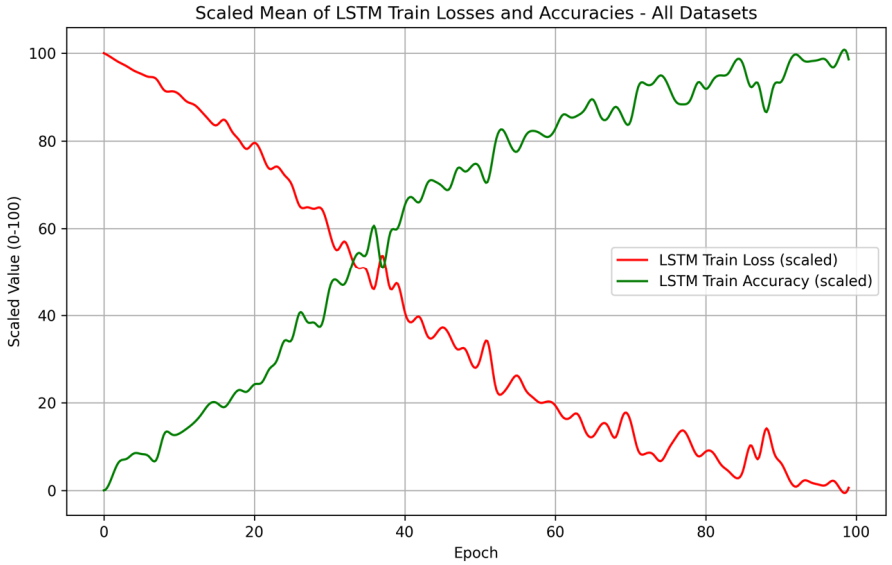


Figure 6. Scaled mean of LSTM training losses and accuracies across all datasets.

5.5. Failure Case Analysis

The evaluation of the proposed method on the KTH and UCF Sports datasets revealed specific instances where the model misclassified certain actions, shedding light on potential limitations in the spatial–temporal feature extraction process. Figures 7 and 8 illustrate these failure cases, demonstrating the complexity of distinguishing between visually or contextually similar actions.

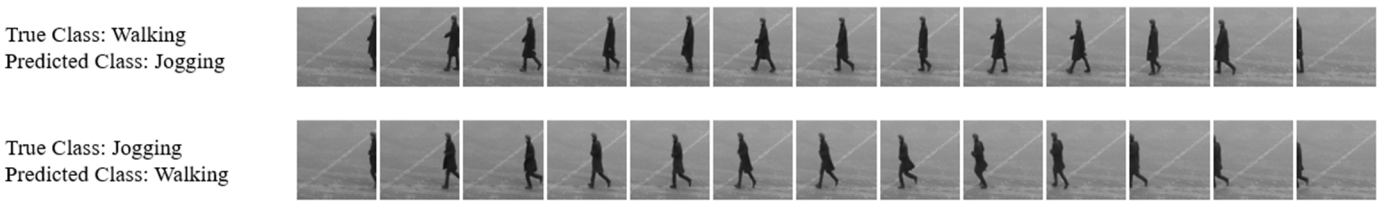


Figure 7. Failure case analysis for the KTH dataset. Examples of misclassifications where the model predicted walking as jogging and jogging as walking, illustrating the challenges in distinguishing between actions with similar motion dynamics and overlapping spatial–temporal features.

Figure 8. Failure case analysis for the UCF Sports dataset. Examples of misclassifications where the model predicted skateboarding as walking and walking as skateboarding, emphasizing the challenges posed by visually complex actions with overlapping spatial features and varied contextual backgrounds.

In the KTH dataset, failure cases primarily occurred between actions such as walking and jogging. Both actions involve similar motion dynamics, with overlapping spatial and temporal features, making them challenging to differentiate. For example, the model incorrectly classified walking as jogging due to the similarity in leg movements and pace, especially in a controlled environment where background and context do not provide additional discriminative cues. Similarly, jogging was sometimes predicted as walking, highlighting the need for improved temporal attention mechanisms to capture subtle motion differences.

For the UCF Sports dataset, failure cases were observed in more dynamic and contextually complex scenarios. As depicted in Figure 8, the model misclassified skateboarding as walking and vice versa. The misclassification of skateboarding as walking could be attributed to the model's inability to fully capture motion-specific patterns, such as the movement of the skateboard or the posture of the subject, especially in sequences where motion blur or background complexity is present. Conversely, walking was occasionally predicted as skateboarding due to the presence of similar body postures in some frames, compounded by the model's potential focus on irrelevant features in noisy or cluttered environments.

6. Discussion

This section presents a comprehensive comparison and evaluation of the performance and efficiency of the proposed method against competing approaches. The analysis is divided into two subsections: the first subsection focuses on comparing the proposed method with competing methods using KTH and UCF Sports datasets, while the second subsection extends the comparison to include the HMDB51 dataset. This structured approach ensures a thorough assessment of the proposed method's effectiveness across different datasets and benchmarks.

6.1. Comparative Analysis of KTH and UCF Sports

For a fair comparison of the KTH and UCF Sports datasets, the experimental implementation conditions, including the selection of training and testing videos, were aligned with the methodologies outlined in [26]. In this evaluation, the performance of the proposed method was assessed using these two challenging action recognition datasets. A leave-one-out testing scheme was employed, as it provides a more robust evaluation compared to conventional data-splitting approaches [17,47]. Under this scheme, all action videos associated with one individual were designated for testing, while videos from the remaining individuals were used for training. This process was repeated iteratively for

each individual, and the average accuracy across all iterations was computed to provide a comprehensive measure of the model's performance.

The accuracy of the competing methods reported in Table 2 was taken directly from their respective papers. These methods were evaluated under different computational architectures, and their computational costs were not always explicitly provided. While we aimed to align experimental setups for fairness, computational environments and resources can inherently vary across studies.

Table 2. Comparison of action recognition accuracy on the KTH dataset with different methods.

Reference	Method	Accuracy (%)
Laptev et al. [48]	Cuboids + HOG3D	91.4
Nazir et al. [49]	3DHarris + 3DSIFT	91.82
Niebles et al. [50]	PLSA	83.33
Jhuang et al. [51]	HMAX	91.70
Taylor et al. [47]	3D GRBM	90
Le et al. [52]	Hierarchical ISA	93.9
Ji et al. [53]	3D CNN	90.2
Sun et al. [54]	3D (DL-SFA)	93.10
Zhang et al. [55]	Dual-channel deep network	92.8
Han et al. [56]	Two-stream ConvNets	93.1
Abdelbaky et al. [26]	ST-VLAD-PCANet	93.33
Chou et al. [57]	NMC	90.58
Liu et al. [58]	EPF + AdaBoost + WLNBNN	94.8
Rodriguez et al. [59]	OneOut + SHMM	94.2
Shi et al. [60]	Final three-stream sDTD	96.8
Proposed method	2D Conv-RBM + LSTM	**97.3**

As shown in Table 2, the proposed method achieved remarkable performance on the KTH dataset, surpassing several state-of-the-art approaches. Traditional methods, such as [48] with Cuboids + HOG3D and [49] with 3DHarris + 3DSIFT, achieved accuracies of 91.4% and 91.82%, respectively, demonstrating the limitations of handcrafted feature extraction techniques. Similarly, methods leveraging probabilistic models, like [50], using PLSA with an accuracy of 83.33%, fell short of modern deep learning methods.

Deep learning-based approaches, such as hierarchical ISA [52] (93.9%), DL-SFA (93.10%), and dual-channel networks [55] (92.8%), showed significant improvement due to their ability to learn hierarchical features. Among these, the sequential trajectory descriptor [60] achieved the highest accuracy of 96.8%, highlighting the effectiveness of multi-stream CNNs in modeling complex action sequences. However, our proposed method outperformed this by leveraging the strengths of 2D Conv-RBM for efficient spatial feature extraction and LSTM for robust temporal modeling, achieving a 97.3% accuracy.

The results indicate that the integration of spatially adaptive feature extraction and temporal dependencies in the proposed model enables it to handle variations in motion patterns and backgrounds more effectively than competing methods. This demonstrates its robustness and generalizability, particularly in controlled environments like the KTH dataset, making it a state-of-the-art solution for human action recognition.

The proposed method achieved an impressive accuracy on the UCF Sports dataset, significantly outperforming many existing approaches. Table 3 presents a comparison of the accuracy between the proposed method and other existing methods. Traditional feature-based methods, such as Dense + HOF [48] and Dense + HOG3D [61], achieved accuracies of 82.6% and 85.6%, respectively, reflecting the limitations of handcrafted feature extraction in handling complex, real-world action dynamics. Similarly, methods leveraging hierarchical feature learning, such as hierarchical ISA [52] and the dual-channel DNN [55],

improved accuracy to 86.5% and 86.7% but still fell short when compared to more recent deep learning-based techniques.

Table 3. Comparison of action recognition accuracy on the UCF Sports dataset with different methods.

Reference	Method	Accuracy (%)
Laptev et al. [48]	Dense + HOF	82.6
Klaser et al. [61]	Dense + HOG3D	85.6
Rahmani et al. [62]	Deep R-NKTM	90
Le et al. [52]	Hierarchical ISA	86.5
Zhang et al. [55]	Dual-channel DNN	86.7
Sun et al. [54]	3D (DL-SFA)	86.6
Yuan et al. [63]	3D Deep model	87.30
Wang et al. [64]	LSTM+CNN	91.89
Ahmed and Aly [65]	STMEI-PCANet	86.7
Abdelbaky et al. [26]	ST(W/o encoding)	90
Proposed method	2D Conv-RBM+LSTM	**94.8**

Deep learning models demonstrated notable improvements in capturing the intricate spatial and temporal dependencies of sports activities. For instance, the 3D deep model [64] achieved 87.3%, and the combination of LSTM and CNN [65] reached 91.89%, showcasing the effectiveness of hybrid approaches in leveraging both spatial and temporal patterns. However, the proposed method surpassed these performances by achieving 94.8% accuracy, demonstrating its superior ability to extract meaningful spatial features through the 2D Conv-RBM while modeling temporal dependencies with the LSTM.

6.2. Comparative Analysis on HMDB51

The HMDB51 dataset offers three distinct splits for training and testing purposes. The final result is determined by averaging the classification results across these splits of training and test data. This dataset presents significant challenges due to its complexity and diversity, including substantial variations in motion dynamics, occlusion, and background noise.

Table 4 provides a comparison of the proposed method with state-of-the-art approaches, highlighting the diversity of methods and their respective performances.

Table 4. Comparison of action recognition accuracy on the HMDB51 dataset with different methods.

Reference	Method	Accuracy (%)
Girdhar et al. [66]	Pose regul. Atten. Pooling	52.2
Meng et al. [67]	Spatio-temporal Atten.	53.1
Du et al. [68]	C3D	46.7
Qiu et al. [69]	P3D-199	62.9
Simonyan et al. [21]	Two-stream CNN	59.4
Wang et al. [70]	TDD	63.2
Feichtenhofer et al. [71]	Two-stream fusion	65.4
Wang et al. [72]	TSN	69.4
Yudistira et al. [73]	TSN Corrnet	70.6
Zong et al. [74]	MSM-ResNets	66.7
Soomro et al. [75]	ARTNet-Res18	70.9
Liu et al. [76]	AMFNet-C	71.2
Du et al. [77]	RSTAN (TSN)	70.5
Liu et al. [27]	STILT	72.1
Chen et al. [78]	AdaptFormer	55.6

Table 4. *Cont.*

Reference	Method	Accuracy (%)
Ranasinghe et al. [79]	SVT(Fine-tune)	67.2
Xing et al. [80]	SVFormer-B	68.2
Khowaja et al. [81]	SINs	83.7
Cong et al. [82]	FNNT	81.24
Kalfaoglu et al. [83]	LTM 3D CNN + BERT	85.1
Wang et al. [84]	VideoMAE V2	**88.1**
Proposed method	2D Conv-RBM+LSTM	81.5

Early methods such as the two-stream CNN [21] achieved 59.4%, and trajectory-pooled deep-convolution descriptors (TDDs) [70] improved this to 63.2%. These approaches primarily leveraged spatial and temporal fusion strategies, which were further refined by models like two-stream fusion [71] (65.4%) and temporal segment networks (TSN) [72] (69.4%). Advanced architectures such as TSN CorrNet [73] (70.6%) and ARTNet-Res18 [75] (70.9%) introduced sophisticated feature aggregation techniques that pushed performance benchmarks further. More recent approaches, including STILT [27] (72.1%) and FNNT [82] (81.24%), incorporated temporal refinement and attention mechanisms, highlighting the growing emphasis on temporal modeling. One of the highest reported accuracy before this study was achieved by LTM 3D CNN + BERT [83], which leveraged a hybrid architecture combining temporal modeling and natural language processing techniques to achieve 85.1%. Additionally, transformer-based methods such as VideoMAE V2 [84], a prominent model in this domain, demonstrated exceptional performance, achieving an accuracy of 88.1% on the HMDB51 dataset. Transformers, with their global attention mechanisms, have proven highly effective in capturing both spatial and temporal dependencies, setting new benchmarks in video action recognition.

The proposed method, based on 2D Conv-RBM and LSTM, achieved a competitive accuracy of 81.5% on HMDB51. While transformer-based models like VideoMAE V2 demonstrated higher performance, they typically require significantly more computational resources and large-scale pretraining on video datasets. In contrast, our method strikes a balance between computational efficiency and accuracy, leveraging the unsupervised feature learning of Conv-RBM and the sequential modeling power of LSTM to handle the dataset's challenges effectively. This positions the proposed approach as a viable and resource-efficient solution for action recognition in complex scenarios.

6.3. Computational Cost Analysis

The proposed method was implemented using two NVIDIA Tesla T4 GPUs with 15,360 MiB of memory each, under CUDA 12.2 with the PyTorch framework. The average inference time per frame was approximately 8 ms with a total training time of 36 h for the KTH dataset, 10 h for the UCF Sports dataset, and 280 h for the HMDB51 dataset. The memory usage per batch during training was 3000 MiB. These metrics provide a practical understanding of the computational demands of the proposed method. While the computational costs of the competing methods are not always reported in the literature, we acknowledge that comparing methods with differing architectures and resource requirements can lead to variability in efficiency and scalability. To address this, our method prioritizes computational efficiency through techniques such as smart frame selection and the use of 2D Conv-RBM for lightweight spatial feature extraction, striking a balance between accuracy and resource consumption.

6.4. Computational Complexity Analysis

To comprehensively evaluate the efficiency of the proposed method, we present a detailed theoretical analysis of the computational complexity for each of its core components: Conv-RBM, LSTM, and the smart frame selection mechanism. This analysis demonstrates the practicality of the proposed approach in terms of temporal and spatial complexities and highlights its suitability for real-world, resource-constrained applications.

6.4.1. Conv-RBM: Spatial Complexity

The 2D Conv-RBM performs convolutional operations to extract spatial features while leveraging weight sharing to reduce computational overhead. Given an input frame of dimensions $H \times W$, a convolutional filter of size $k \times k$, and F feature maps, the spatial complexity for a single convolutional layer is

$$O_{Conv-RBM} = H \times W \times k^2 \times F, \tag{20}$$

This complexity arises from convolving each input pixel with the kernel over F feature maps. Compared to standard RBMs, which rely on fully connected operations, the Conv-RBM significantly reduces the parameter count due to the shared weights in the convolutional operation [34]. Additionally, pooling layers, which follow convolution, reduce the spatial dimensions by a factor of $s \times s$ (e.g., for 2×2 pooling, $s = 2$), further optimizing computational cost.

6.4.2. LSTM: Temporal Complexity

The LSTM processes sequences of spatial features extracted by the Conv-RBM to capture temporal dependencies. For a video sequence of length T, with each feature vector having a dimension d and assuming h hidden units in the LSTM, the temporal complexity for a single layer is

$$O_{LSTM} = T \times \left(4 \times h^2 + 4 \times h \times d\right), \tag{21}$$

The factor of 4 arises from the gating mechanisms (input, forget, cell state, and output gates) inherent to the LSTM architecture [85]. The complexity scales linearly with the sequence length T, making the LSTM computationally efficient for moderately sized sequences, as is typical in action recognition tasks. While transformer-based methods such as VideoMAE V2 have higher representational capacity, they exhibit quadratic complexity with respect to T, making LSTMs more practical for resource-constrained environments.

6.4.3. Smart Frame Selection Mechanism

The frame selection mechanism operates in two stages: single-frame evaluation and global evaluation. For a video sequence of T frames, the single-frame selector computes a confidence score δi for each frame, with complexity proportional to T:

$$O_{SingleFrameSelector} = T \times d_{MLP}, \tag{22}$$

where d_{MLP} is the complexity of the lightweight multi-layer perceptron used for score computation. The global selector evaluates pairs of frames using an attention and relational network, which requires concatenating T frames with T_r randomly selected subsequent frames, leading to complexity:

$$O_{GlobalSelector} = T \times T_r \times d_{Attention}, \tag{23}$$

where $d_{Attention}$ represents the complexity of the attention mechanism. As T_r is typically much smaller than T, this operation remains efficient while capturing key temporal relationships [38].

6.4.4. Overall Complexity

The total complexity of the proposed method is a summation of the complexities of its components:

$$O_{Total} = O_{Conv-RBM} + O_{LSTM} + O_{FrameSelection}, \quad (24)$$

By reducing the number of frames processed through smart frame selection and employing parameter-efficient Conv-RBM layers, the proposed method achieves a balance between computational efficiency and performance. This lightweight design contrasts with more resource-intensive approaches, such as transformer-based models, which exhibit higher spatial and temporal complexity.

6.5. Limitations and Future Directions

While the proposed 2D Conv-RBM + LSTM architecture demonstrates promising performance in video-based human action recognition tasks, several limitations warrant further exploration to enhance its applicability and robustness.

- Computational Trade-offs:

Despite the lightweight design of the 2D Conv-RBM and the use of smart frame selection, the overall computational requirements remain non-trivial, particularly for larger datasets with higher-resolution videos. This could limit the scalability of the model for real-time applications or scenarios with constrained hardware resources. Future work could explore further optimization techniques, such as quantization or pruning, to reduce the computational overhead.

- Single-layer Conv-RBM Design:

The proposed method employs a single-layer Conv-RBM for spatial feature extraction. While this design emphasizes computational efficiency, deeper architectures could provide richer hierarchical features and better representations. Future studies could investigate the trade-offs of stacking multiple Conv-RBM layers or integrating multi-scale feature extraction.

- Temporal Modeling with LSTM:

Although LSTM networks effectively capture temporal dependencies, they may struggle with very long sequences or subtle temporal dynamics. Exploring alternative temporal modeling approaches, such as attention-based mechanisms or transformer architectures, could enhance the temporal learning capabilities of the model.

- Failure Cases and Error Patterns:

As discussed in the failure case analysis, the model occasionally struggles with distinguishing visually similar actions, particularly in scenarios with high motion blur or background clutter. Enhancing spatial–temporal attention mechanisms or incorporating multi-modal inputs (e.g., depth or optical flow) could help mitigate these issues.

7. Conclusions

The analysis of the comparison tables and the experimental results highlights the efficacy and robustness of the proposed 2D Conv-RBM+LSTM architecture for video-based human action recognition. By integrating spatial feature extraction through convolutional restricted Boltzmann machines and temporal modeling via LSTMs, the model demonstrated

superior performance across multiple challenging datasets, including KTH, UCF Sports, and HMDB51. The proposed method consistently outperformed traditional handcrafted approaches and several state-of-the-art deep learning architectures, achieving notable accuracy improvements. This study has significant practical implications for various domains requiring robust action recognition capabilities. The lightweight and efficient design of the proposed model makes it well suited for real-time applications such as surveillance systems, where computational efficiency is critical for monitoring dynamic environments. Similarly, the model's ability to accurately capture complex motion patterns has potential applications in sports analytics, enabling detailed performance assessments and tactic evaluations. Furthermore, the system's robustness positions it as a valuable tool in healthcare monitoring for activities such as patient rehabilitation and elderly care. In human–computer interaction, the proposed approach can enhance gesture recognition and virtual reality applications by delivering real-time and accurate action recognition.

The smart frame selection mechanism significantly reduces computational overhead, making the method scalable for deployment on resource-constrained devices. The modular nature of the architecture ensures its adaptability to diverse datasets and environmental conditions, with potential extensions involving multi-modal data integration to further improve performance. While the current study emphasizes computational efficiency and generalizability, future work will explore deeper architectures and alternative temporal modeling approaches, such as transformer-based frameworks, to address challenges posed by more complex video datasets. In conclusion, the proposed hybrid architecture achieves a balanced trade-off between accuracy and computational efficiency, demonstrating its viability for a wide range of video-based action recognition tasks. These findings underscore the importance of efficient spatial–temporal modeling and provide a foundation for further advancements in the field of human action recognition.

Author Contributions: Conceptualization, M.J. and M.I.; methodology, M.J.; software, M.J. and M.I.; validation, M.J., M.I. and H.R.A.; formal analysis, M.J.; investigation, M.J.; resources, M.J.; data curation, M.I.; writing—original draft preparation, M.J.; writing—review and editing, M.J. and H.R.A.; visualization, M.I.; supervision, H.R.A.; project administration, M.J.; All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original data presented in the study are openly available: KTH: https://www.csc.kth.se/cvap/actions/ (accessed on 10 February 2024), UCF Sports: https://www.crcv.ucf.edu/research/data-sets/ucf-sports-action/ (accessed on 16 April 2024), and HMDB51: https://serre-lab.clps.brown.edu/resource/hmdb-a-large-human-motion-database/ (accessed on 22 May 2024).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Mihanpour, A.; Rashti, M.J.; Alavi, S.E. Human Action Recognition in Video Using DB-LSTM and ResNet. In Proceedings of the 2020 IEEE International Conference on Wireless Research (ICWR), Tehran, Iran, 22 April 2020. [CrossRef]
2. Ma, M.; Marturi, N.; Li, Y.; Leonardis, A.; Stolkin, R. Region-Sequence Based Six-Stream CNN Features for General and Fine-Grained Human Action Recognition in Videos. *Pattern Recognit.* **2018**, *76*, 545–558. [CrossRef]
3. Dai, C.; Liu, X.; Zhong, L.; Yu, T. Video-Based Action Recognition Using Spatial and Temporal Features. In Proceedings of the 2018 IEEE Cybermatics Congress, Halifax, NS, Canada, 30 July 2018. [CrossRef]
4. Johnson, D.R.; Uthariaraj, V.R. A Novel Parameter Initialization Technique Using RBM-NN for Human Action Recognition. *Comput. Intell. Neurosci.* **2020**, *1*, 30. [CrossRef]

5. Cob-Parro, A.C.; Losada-Gutiérrez, C.; Marrón Romera, M.; Gardel Vicente, A.; Muñoz, I.B. A New Framework for Deep Learning Video-Based Human Action Recognition on the Edge. *Expert Syst. Appl.* **2023**, *238*, 122220. [CrossRef]
6. Silva, D.; Manzo-Martinez, A.; Gaxiola, F.; Gonzales-Gurrola, L.C.; Alonso, G.R. Analysis of CNN Architectures for Human Action Recognition in Video. *Comput. Sist.* **2022**, *26*, 67–80. [CrossRef]
7. Manakitsa, N.; Maraslidis, G.S.; Moysis, L.; Fragulis, G.F. A Review of Machine Learning and Deep Learning for Object Detection, Semantic Segmentation, and Human Action Recognition in Machine and Robotic Vision. *Technologies* **2024**, *12*, 15. [CrossRef]
8. Soentanto, P.N.; Hendryli, J.; Herwindiati, D. Object and Human Action Recognition from Video Using Deep Learning Models. In Proceedings of the 6th International Conference on Signal and Image Processing Systems (ICSIGSYS), Bandung, Indonesia, 16 July 2019; pp. 88–93. [CrossRef]
9. Begampure, S.; Jadhav, P.M. Intelligent Video Analytics for Human Action Detection: A Deep Learning Approach with Transfer Learning. *Int. J. Comput. Dig. Syst.* **2022**, *11*, 57–72. [CrossRef]
10. Li, C.; Huang, Q.; Li, X.; Wu, Q. Human Action Recognition Based on Multi-Scale Feature Maps from Depth Video Sequences. *Multimed. Tools Appl.* **2021**, *80*, 32111–32130. [CrossRef]
11. Liu, X.; Yang, X. Multi-Stream with Deep Convolutional Neural Networks for Human Action Recognition in Videos. In *Neural Information Processing, Proceedings of the 25th International Conference, ICONIP 2018, Siem Reap, Cambodia, 13 December 2018*; Proceedings, Part I 25; Springer International Publishing: Cham, Switzerland, 2018; pp. 251–262. [CrossRef]
12. Ulhaq, A.; Akhtar, N.; Pogrebna, G.; Mian, A. Vision Transformers for Action Recognition: A Survey. *arXiv* **2022**, arXiv:2209.05700.
13. Dosovitskiy, A. An Image Is Worth 16×16 Words: Transformers for Image Recognition at Scale. *arXiv* **2020**, arXiv:2010.11929.
14. Arnab, A.; Dehghani, M.; Heigold, G.; Sun, C.; Lučić, M.; Schmid, C. ViViT: A Video Vision Transformer. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 11 October 2021; pp. 6836–6846.
15. Bertasius, G.; Wang, H.; Torresani, L. Is Space-Time Attention All You Need for Video Understanding? In Proceedings of the 38th International Conference on Machine Learning (ICML), Virtual, 18 July 2021; Volume 2, p. 4.
16. Khan, S.; Naseer, M.; Hayat, M.; Zamir, S.W.; Khan, F.S.; Shah, M. Transformers in Vision: A Survey. *ACM Comput. Surv.* **2022**, *54*, 1–41. [CrossRef]
17. Schüldt, C.; Laptev, I.; Caputo, B. Recognizing Human Actions: A Local SVM Approach. In Proceedings of the 17th International Conference on Pattern Recognition (ICPR), Cambridge, UK, 23 August 2004; pp. 32–36. [CrossRef]
18. Rodriguez, M.D.; Ahmed, J.; Shah, M. Action mach a spatio-temporal maximum average correlation height filter for action recognition. In Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 23 June 2008.
19. Kuehne, H.; Jhuang, H.; Garrote, E.; Poggio, T.; Serre, T. HMDB: A Large Video Database for Human Motion Recognition. In Proceedings of the 2011 International Conference on Computer Vision (ICCV), Barcelona, Spain, 6 November 2011. [CrossRef]
20. Raza, A.; Al Nasar, M.R.; Hanandeh, E.S.; Zitar, R.A.; Nasereddin, A.Y.; Abualigah, L. A Novel Methodology for Human Kinematics Motion Detection Based on Smartphones Sensor Data Using Artificial Intelligence. *Technologies* **2023**, *11*, 55. [CrossRef]
21. Simonyan, K.; Zisserman, A. Two-Stream Convolutional Networks for Action Recognition in Videos. In *Advances in Neural Information Processing Systems (NeurIPS)*; Curran Associates, Inc.: Newry, UK, 2014; Volume 27, pp. 568–576.
22. Zhu, J.; Zou, W.; Zhu, Z.; Xu, L.; Huang, G. Action Machine: Toward Person-Centric Action Recognition in Videos. *IEEE Signal Process. Lett.* **2019**, *11*, 1633–1637. [CrossRef]
23. Carreira, J.; Zisserman, A. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21 July 2017. [CrossRef]
24. Kulkarni, S.S.; Jadhav, S. Insight on Human Activity Recognition Using the Deep Learning Approach. In Proceedings of the International Conference on Emerging Smart Computing and Informatics (ESCI), Pune, India, 1 March 2023. [CrossRef]
25. Yang, C.; Mei, F.; Zang, T.; Tu, J.; Jiang, N.; Liu, L. Human Action Recognition Using Key-Frame Attention-Based LSTM Networks. *Electronics* **2023**, *12*, 2622. [CrossRef]
26. Abdelbaky, A.; Aly, S. Two-Stream Spatiotemporal Feature Fusion for Human Action Recognition. *Vis. Comput.* **2021**, *37*, 1821–1835. [CrossRef]
27. Liu, T.; Ma, Y.; Yang, W.; Ji, W.; Wang, R.; Jiang, P. Spatial-Temporal Interaction Learning Based Two-Stream Network for Action Recognition. *Inf. Sci.* **2022**, *606*, 864–876. [CrossRef]
28. Donahue, J.; Anne Hendricks, L.; Guadarrama, S.; Rohrbach, M.; Venugopalan, S.; Saenko, K.; Darrell, T. Long-Term Recurrent Convolutional Networks for Visual Recognition and Description. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7 June 2015; pp. 2625–2634. [CrossRef]
29. Zheng, T.; Liu, C.; Liu, B.; Wang, M.; Li, Y.; Wang, P.; Qin, X.; Guo, Y. Scene Recognition Model in Underground Mines Based on CNN-LSTM and Spatial-Temporal Attention Mechanism. In Proceedings of the 2020 International Symposium on Computer, Consumer, and Control (IS3C), Taichung City, Taiwan, 13 November 2020; pp. 513–516. [CrossRef]
30. Saoudi, E.M.; Jaafari, J.; Andaloussi, S.J. Advancing Human Action Recognition: A Hybrid Approach Using Attention-Based LSTM and 3D CNN. *Sci. Afr.* **2023**, *21*, e01796. [CrossRef]

31. Liu, D.; Yan, Y.; Shyu, M.; Zhao, G.; Chen, M. Spatio-Temporal Analysis for Human Action Detection and Recognition in Uncontrolled Environments. *Int. J. Multimed. Data Eng. Manag. (IJMDEM)* **2015**, *1*, 1–18. [CrossRef]
32. Su, Y. Implementation and Rehabilitation Application of Sports Medical Deep Learning Model Driven by Big Data. *IEEE Access* **2019**, *7*, 156338–156348. [CrossRef]
33. Hossen, M.A.; Naim, A.G.; Abbas, P.E. Deep Learning for Skeleton-Based Human Activity Segmentation: An Autoencoder Approach. *Technologies* **2024**, *12*, 96. [CrossRef]
34. Lee, H.; Grosse, R.; Ranganath, R.; Ng, A.Y. Convolutional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical Representations. In Proceedings of the 26th Annual International Conference on Machine Learning, Montreal, QC, Canada, 14 June 2009; pp. 609–616.
35. Osadchy, M.; Miller, M.; Cun, Y. Synergistic Face Detection and Pose Estimation with Energy-Based Models. In *Advances in Neural Information Processing Systems 17*; MIT Press: Cambridge, MA, USA, 2005.
36. LeCun, Y.; Bengio, Y.; Hinton, G. Deep Learning. *Nature* **2015**, *521*, 436–444. [CrossRef]
37. Salakhutdinov, R.; Hinton, G. Deep Boltzmann Machines. In Proceedings of the 12th International Conference on Artificial Intelligence and Statistics (AISTATS), Clearwater Beach, FL, USA, 16 April 2009; pp. 448–455.
38. Gowda, S.N.; Rohrbach, M.; Sevilla-Lara, L. Smart Frame Selection for Action Recognition. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual, 2 February 2021; Volume 35, pp. 1451–1459. [CrossRef]
39. Zhang, X.; Liu, T.; Lo, K.; Feng, J. Dynamic Selection and Effective Compression of Key Frames for Video Abstraction. *Pattern Recognit. Lett.* **2003**, *24*, 1523–1532. [CrossRef]
40. Hasebe, S.; Nagumo, M.; Muramatsu, S.; Kikuchi, H. Video Key Frame Selection by Clustering Wavelet Coefficients. In Proceedings of the 12th European Signal Processing Conference (EUSIPCO), Vienna, Austria, 6 September 2004. [CrossRef]
41. Xu, Q.; Wang, P.; Long, B.; Sbert, M.; Feixas, M.; Scopigno, R. Selection and 3D Visualization of Video Key Frames. In Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (ICSMC), Istanbul, Turkey, 10 October 2010. [CrossRef]
42. Kulbacki, M.; Segen, J.; Chaczko, Z.; Rozenblit, J.; Klempous, R.; Wojciechowski, K. Intelligent Video Analytics for Human Action Recognition: The State of Knowledge. *Sensors* **2023**, *9*, 4258. [CrossRef] [PubMed]
43. Feichtenhofer, C.; Pinz, A.; Wildes, R.P.; Zisserman, A. Deep Insights into Convolutional Networks for Video Recognition. *Int. J. Comput. Vis.* **2020**, *128*, 420–437. [CrossRef]
44. Tsai, J.K.; Hsu, C.; Wang, W.Y.; Huang, S.K. Deep Learning-Based Real-Time Multiple-Person Action Recognition System. *Sensors* **2020**, *17*, 4857. [CrossRef]
45. Fischer, A.; Igel, C. An Introduction to Restricted Boltzmann Machines. In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, Proceedings of the 17th Iberoamerican Congress, CIARP, Buenos Aires, Argentina, 3–6 September 2012*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 14–36. [CrossRef]
46. Srivastava, N.; Salakhutdinov, R. Multimodal Learning with Deep Boltzmann Machines. *J. Mach. Learn. Res.* **2012**, *15*, 2949–2980.
47. Taylor, G.W.; Fergus, R.; LeCun, Y.; Bregler, C. Convolutional Learning of Spatio-Temporal Features. In Proceedings of the 11th European Conference on Computer Vision, Heraklion, Greece, 5 September 2010. [CrossRef]
48. Laptev, I.; Marszałek, M.; Schmid, C.; Rozenfeld, B. Learning Realistic Human Actions from Movies. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Anchorage, AK, USA, 23 June 2008. [CrossRef]
49. Nazir, S.; Yousaf, M.H.; Velastin, S.A. Evaluating a Bag-of-Visual Features Approach Using Spatio-Temporal Features for Action Recognition. *Comput. Electr. Eng.* **2018**, *72*, 660–669. [CrossRef]
50. Niebles, J.C.; Wang, H.; Fei-Fei, L. Unsupervised Learning of Human Action Categories Using Spatial-Temporal Words. *Int. J. Comput. Vis.* **2008**, *79*, 299–318. [CrossRef]
51. Jhuang, H.; Serre, T.; Wolf, L.; Poggio, T. A Biologically Inspired System for Action Recognition. In Proceedings of the IEEE 11th International Conference on Computer Vision (ICCV), Rio de Janeiro, Brazil, 14 October 2007. [CrossRef]
52. Le, Q.V.; Zou, W.Y.; Yeung, S.Y.; Ng, A.Y. Learning Hierarchical Invariant Spatio-Temporal Features for Action Recognition with Independent Subspace Analysis. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Colorado, CO, USA, 20 June 2011. [CrossRef]
53. Ji, S.; Xu, W.; Yang, M.; Yu, K. 3D Convolutional Neural Networks for Human Action Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 221–231. [CrossRef]
54. Sun, L.; Jia, K.; Chan, T.H.; Fang, Y.; Wang, G.; Yan, S. DL-SFA: Deeply-Learned Slow Feature Analysis for Action Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23 June 2014. [CrossRef]
55. Zhang, K.; Zhang, L. Extracting Hierarchical Spatial and Temporal Features for Human Action Recognition. *Multimed. Tools Appl.* **2018**, *77*, 16053–16068. [CrossRef]
56. Han, Y.; Zhang, P.; Zhuo, T.; Huang, W.; Zhang, Y. Going Deeper with Two-Stream ConvNets for Action Recognition in Video Surveillance. *Pattern Recognit. Lett.* **2018**, *107*, 83–90. [CrossRef]

57. Chou, K.P.; Prasad, M.; Wu, D.; Sharma, N.; Li, D.L.; Lin, Y.F.; Blumenstein, M.; Lin, W.C.; Lin, C.T. Robust Feature-Based Automated Multi-View Human Action Recognition System. *IEEE Access* **2018**, *6*, 15283–15296. [CrossRef]
58. Liu, L.; Shao, L.; Zhen, X.; Li, X. Learning Discriminative Key Poses for Action Recognition. *IEEE Trans. Cybern.* **2013**, *43*, 1860–1870. [CrossRef]
59. Rodriguez, M.; Orrite, C.; Medrano, C.; Makris, D. One-Shot Learning of Human Activity with an MAP Adapted GMM and Simplex-HMM. *IEEE Trans. Cybern.* **2016**, *47*, 1769–1780. [CrossRef]
60. Shi, Y.; Tian, Y.; Wang, Y.; Huang, T. Sequential Deep Trajectory Descriptor for Action Recognition with Three-Stream CNN. *IEEE Trans. Multimed.* **2017**, *19*, 1510–1520. [CrossRef]
61. Klaser, A.; Marszałek, M.; Schmid, C. A Spatio-Temporal Descriptor Based on 3D-Gradients. In Proceedings of the 19th British Machine Vision Conference (BMVC), Leeds, UK, 7 September 2009.
62. Rahmani, H.; Mian, A.; Shah, M. Learning a Deep Model for Human Action Recognition from Novel Viewpoints. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 667–681. [CrossRef]
63. Yuan, C.; Li, X.; Hu, W.; Ling, H.; Maybank, S. 3D R-Transform on Spatio-Temporal Interest Points for Action Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Portland, OR, USA, 25 June 2013; pp. 724–730. [CrossRef]
64. Wang, L.; Xu, Y.; Cheng, J.; Xia, H.; Yin, J.; Wu, J. Human Action Recognition by Learning Spatio-Temporal Features with Deep Neural Networks. *IEEE Access* **2018**, *6*, 17913–17922. [CrossRef]
65. Ahmed, A.; Aly, S. Human Action Recognition Using Short-Time Motion Energy Template Images and PCANet Features. *Neural Comput. Appl.* **2020**, *16*, 12561–12574. [CrossRef]
66. Girdhar, R.; Deva, R. Attentional Pooling for Action Recognition. In *Advances in Neural Information Processing Systems (NeurIPS)*; Curran Associates, Inc.: Newry, UK, 2017; Volume 30.
67. Meng, L.; Zhao, B.; Chang, B.; Huang, G.; Sun, W.; Tung, F.; Sigal, L. Interpretable Spatio-Temporal Attention for Video Action Recognition. In Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCV), Seoul, Republic of Korea, 27 October 2019. [CrossRef]
68. Du, T.; Bourdev, L.; Fergus, R. Learning Spatiotemporal Features with 3D Convolutional Networks. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7 December 2015. [CrossRef]
69. Qiu, Z.; Yao, T.; Mei, T. Learning Spatio-Temporal Representation with Pseudo-3D Residual Networks. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22 October 2017. [CrossRef]
70. Wang, L.; Qiao, Y.; Tang, X. Action Recognition with Trajectory-Pooled Deep-Convolutional Descriptors. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7 June 2015. [CrossRef]
71. Feichtenhofer, C.; Pinz, A.; Zisserman, A. Convolutional Two-Stream Network Fusion for Video Action Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27 June 2016. [CrossRef]
72. Wang, L.; Yuan, X.; Zhe, W.; Qiao, Y.; Lin, D.; Tang, X.; Van Gool, L. Temporal Segment Networks: Towards Good Practices for Deep Action Recognition. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 11 October 2016. [CrossRef]
73. Yudistira, N.; Kurita, T. Correlation Net: Spatiotemporal Multimodal Deep Learning for Action Recognition. *Signal Process. Image Commun.* **2020**, *82*, 115731. [CrossRef]
74. Zong, M.; Wang, R.; Chen, X. Motion Saliency Based Multi-Stream Multiplier ResNets for Action Recognition. *Image Vis. Comput.* **2021**, *107*, 104108. [CrossRef]
75. Soomro, K.; Zamir, A.R.; Shah, M. UCF101: A Dataset of 101 Human Actions Classes from Videos in the Wild. *arXiv* **2012**, arXiv:1212.0402.
76. Liu, S.; Ma, X. Attention-Driven Appearance-Motion Fusion Network for Action Recognition. *IEEE Trans. Multimed.* **2022**, *25*, 2573–2584. [CrossRef]
77. Du, W.; Wang, Y.; Qiao, Y. Recurrent Spatial-Temporal Attention Network for Action Recognition in Videos. *IEEE Trans. Image Process.* **2018**, *27*, 1347–1360. [CrossRef] [PubMed]
78. Chen, S.; Ge, C.; Tong, Z.; Wang, J.; Song, Y.; Wang, J.; Luo, P. AdaptFormer: Adapting Vision Transformers for Scalable Visual Recognition. In *Advances in Neural Information Processing Systems (NeurIPS)*; Curran Associates, Inc.: Newry, UK, 2022; Volume 35, pp. 16664–16678.
79. Ranasinghe, K.; Naseer, M.; Khan, S.; Khan, F.S.; Ryoo, M.S. Self-Supervised Video Transformer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18 June 2022. [CrossRef]
80. Xing, Z.; Dai, Q.; Hu, H.; Chen, J.; Wu, Z.; Jiang, Y.G. SvFormer: Semi-Supervised Video Transformer for Action Recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 18 June 2023; pp. 18816–18826.

81. Khowaja, S.A.; Lee, S.L. Semantic Image Networks for Human Action Recognition. *Int. J. Comput. Vis.* **2020**, *128*, 393–419. [CrossRef]
82. Cong, G.; Domeniconi, G.; Yang, C.C.; Shapiro, J.; Zhou, F.; Chen, B. Fast Neural Network Training on a Cluster of GPUs for Action Recognition with High Accuracy. *J. Parallel Distrib. Comput.* **2019**, *134*, 153–165. [CrossRef]
83. Kalfaoglu, M.E.; Kalkan, S.; Alatan, A.A. Late Temporal Modeling in 3D CNN Architectures with BERT for Action Recognition. In *Computer Vision—ECCV 2020 Workshops, Proceedings of the 16th European Conference on Computer Vision, Glasgow, UK, 23 August 2020*; Springer International Publishing: Cham, Switzerland, 2020; pp. 731–747. [CrossRef]
84. Wang, L.; Huang, B.; Zhao, Z.; Tong, Z.; He, Y.; Wang, Y.; Wang, Y.; Qia, Y. Videomae v2: Scaling video masked autoencoders with dual masking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18 June 2023; pp. 14549–14560.
85. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]