modelling

MDPI

Article

Human Action Recognition from Videos Using Motion History Mapping and Orientation Based Three-Dimensional Convolutional Neural Network Approach

Ishita Arora [1,*] and M. Gangadharappa [2]

[1] Department of Electronics & Communication, AIACTR, Affiliated to GGSIPU, Delhi 110031, India
[2] Department of Electronics & Communication, NSUT East Campus, Delhi 110031, India; gangadharaccess@gmail.com or m.gangadharappa@nsut.ac.in
* Correspondence: ishitaarora202@gmail.com

Abstract: Human Activity Recognition (HAR) has recently attracted the attention of researchers. Human behavior and human intention are driving the intensification of HAR research rapidly. This paper proposes a novel Motion History Mapping (MHI) and Orientation-based Convolutional Neural Network (CNN) framework for action recognition and classification using Machine Learning. The proposed method extracts oriented rectangular patches over the entire human body to represent the human pose in an action sequence. This distribution is represented by a spatially oriented histogram. The frames were trained with a 3D Convolution Neural Network model, thus saving time and increasing the Classification Correction Rate (CCR). The K-Nearest Neighbor (KNN) algorithm is used for the classification of human actions. The uniqueness of our model lies in the combination of Motion History Mapping approach with an Orientation-based 3D CNN, thereby enhancing precision. The proposed method is demonstrated to be effective using four widely used and challenging datasets. A comparison of the proposed method's performance with current state-of-the-art methods finds that its Classification Correction Rate is higher than that of the existing methods. Our model's CCRs are 92.91%, 98.88%, 87.97.% and 87.77% which are remarkably higher than the existing techniques for KTH, Weizmann, UT-Tower and YouTube datasets, respectively. Thus, our model significantly outperforms the existing models in the literature.

Keywords: action recognition; machine learning; 3D convolution; convolutional neural network (CNN); global histogramming; k-nearest neighbor (KNN)

check for updates

Academic Editor: Wei Gao

Received: 10 March 2025
Revised: 12 April 2025
Accepted: 14 April 2025
Published: 18 April 2025

Citation: Arora, I.; Gangadharappa, M. Human Action Recognition from Videos Using Motion History Mapping and Orientation Based Three-Dimensional Convolutional Neural Network Approach. *Modelling* **2025**, *6*, 33. https://doi.org/10.3390/modelling6020033

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The number of videos uploaded to the internet has increased dramatically over the last few decades. As a consequence, visual content [1] is becoming more prevalent on social media platforms. More than 500 h of video content is uploaded every minute on YouTube. Academic and industrial researchers are developing more proficient techniques for video processing, such as video summarization [2]. Computer vision applications that recognize actions in the video such as video surveillance and anomaly detection are in high demand. A technique for extracting action from video frames is called action segmentation. In various applications, it is used as part of visual effects in movies, to understand scenes in detail, to create virtual backgrounds, and to identify human action automatically in videos without interference from objects.

Nowadays, technology, such as that used for activity motion identification, is more focused and attractive due to advancements in the field of computer vision, as well as in Machine Learning, and finds applications in several areas such as in sports motion analysis, the prediction of pedestrian behavior, fitness tracking systems, human–computer interfaces and many others. Human activity can be determined from three-dimensional images and videos using algorithms developed specifically for this purpose [3,4]. A wide range of applications can be achieved by recognizing human actions. Even after several decades of research and numerous encouraging advancements, identifying human actions remains challenging. The movement of a human is articulated by nature. It is very challenging to extract motions with such high levels of articulation from monocular video sensors. According to studies conducted in the past decade, video-based Human Action Recognition is limited by this difficulty [5].

To detect human activity, texture information such as spatial–temporal gradients [6], spatio-temporal frequency [7,8] and dynamic texture recognition [9] have been considered adequate. Meanwhile, other groups have analyzed motion features by using optical flows directly, such as clustered motion patterns [10,11], motion heat maps [12], crowd prediction [13], spatial saliency with the motion feature [14], particle trajectory, optical flow fields [15], and local motion histogram [16].

Despite significant efforts made by the computer vision and image processing communities, video action recognition remains a challenging problem due to scene complexity, acquisition complexity, and action complexity. One of the primary challenges is to model actions in video sequences, known as action representation. Dynamic models have been proven to be the most effective among many action representations [17], shape models [18], interest-point models [19], and motion models [20].

Computer-based systems have great potential for recognizing human action or posture, but CNN-based techniques have improved model performance. Motion History Mapping (MHI) is an important technique in which the sequence of the silhouette is converted into a set of gray-scale images while preserving the dominant information of motion. Therefore, it has the major advantage of not being much sensitive to silhouette noises such as shadows and missing parts. MHI expresses the motion flow by using the intensity of every pixel in a temporal manner. Moreover, MHI can be implemented using low-powered CPUs. The ordinary and standard Convolutional Neural Network cannot handle rotational invariance. Rotational invariance plays a very crucial role in Human Activity Recognition. The Orientation-based Convolutional Neural Network (CNN) learns features that are invariant to a rotational aspect.

We propose a novel model in this study to achieve improved performance. The key contributions of our work are listed below.

1. In this paper, we have proposed a novel Motion History Mapping and Orientation-based Convolutional Neural Network (CNN) framework for Human Activity Recognition and classification using Machine Learning.
2. A KNN-based classification technique is introduced to classify human activities.
3. The exposure frames were trained with the Orientation-based 3D Convolutional Neural Network (CNN) model, thus increasing the Classification Correction Rate (CCR).
4. This model has used four publicly available and challenging datasets containing movements of human activities.
5. The proposed technique outperforms existing techniques used on the four publicly available and challenging datasets, in terms of higher precision and CCR. Therefore, it can be very advantageous for several accuracy-centric applications.
6. The proposed model is also computationally efficient.

7. Our model performs well even on low-resolution datasets such as UT-Tower, unlike the MediaPipe approach, which is generally not preferable for dark, blurry and noisy low-quality video datasets such as UT-Tower.

Paper Organization

Detailed information about existing research on video processing and Human Action Recognition is presented in Section 2. The four steps in the proposed methodology (viz. rectangle extraction, Histograms of Oriented Rectangles (HORs), training and testing frame sequences, and the classification of K-Nearest Neighbors (KNNs)) are explained in Section 3. Section 4 describes the experimental setup, datasets, results analysis, and discussion to demonstrate the validated findings. Section 5 concludes the proposed work and discusses future work.

2. Literature Review

Research in computer vision and pattern recognition has focused on detecting and recognizing human actions in video sequences for decades [21,22]. Video-salient features are usually extended to video using $(2D + t)$ images combined to form a 3D frame. Feature extraction models are categorized into two categories, namely 2D feature extraction and 3D feature extraction.

(i) 2D Feature Extraction Models

Detectors of 2D features select mainly salient structures in the image, while detectors of 3D features use structured measurements. Using 3D Gaussian filtering, saliency criteria were computed for a 3D autocorrelation matrix [23].

The time domain is not treated similarly to space, and few methods can detect salient features based on motion. These surveys provide detailed information on Human Action Recognition [24–26]. The authors use a taxonomy of 2D and 3D feature detectors and recognition methods [27–30].

Deep learning models [31,32] build high-level features from low-level ones based on learning a hierarchy of features from low-level ones. The Machine Learning (ML) technique is composed of two types—supervised and unsupervised—and is used to train the dataset. The resulting systems can engage in recognizing visual objects competitively [33], Human Action Recognition, Natural Language Processing (NLP) [34], human tracking [35], segmentation tasks [36,37] and denoising [38,39].

The authors of [40] proposed a wearable sensor-based HAR and SA-GAN, a novel method for adversarial knowledge transfer. The concept of spatial–temporal interest points (e.g., cuboids) has become very popular in action recognition [41], 3D Hessians, 3D Harris corners, and 3D salient points [42]. They are mostly extensions of traditional 2D methods used for object detection, which follow a traditional approach for object detection. A codebook is created by computing multiple-scale feature descriptors, clustering them, and assigning them to a Bag-of-Words (BoW) representation. Direct models of spatio–temporal relationships between features have also been proposed [43]. The author of [44] presented a Spatio-Temporal Manifold Network (STMN) system that considered the overfitting problem and used deep feature extraction for feature extraction. Graded algorithms could easily be reversed to address this network's inefficient processing order. Using independent dictionaries and classifiers, a novel framework is presented by the authors [45].

(ii) 3D Feature Extraction Models

An arbitrary view can be identified with the resulting classification algorithm using a dictionary constructed from a real-world 2D image sequence. Three-dimensional images were also enhanced with dense 3D features. The 3D descriptor was introduced by [46]

to recognize activity features. Over the past decade, it has been widely accepted that 2D Human Action Recognition is the most popular form; however, 3D Human Action Recognition is still underdeveloped. Three-dimensional Human Action Recognition has been reported by relatively few authors so far. A 3D environment is established by humans in real-time, unlike traditional cameras that create a 2D scene projection. A survey on pose estimation techniques divides them into 2D and 3D approaches based on achieving a 2D or 3D pose representation [47]. Two-dimensional activities in the image plane can only be analyzed visually because they are projections. Due to this, the projection of actions may not include all information about what was done, depending on the viewpoint. It is possible to resolve this shortcoming by constructing three-dimensional representations based on reconstructed 3D data with two or more cameras. The study by [48] stands out as the sole endeavor to estimate 3D poses directly from individual images. It is also possible to predict the pose of 3D objects by predicting the depth of key points in 2D [49]. LSTMs have been used to predict 3D poses based on single images [50]. Transformer-based models [51] can be used for prediction when dealing with complex and long-range dependency on sensor data, as they follow a parallel processing mechanism. They are computationally expensive to deploy for real-time applications. Temporal Graph Attention Networks (TGATs) [52] use attention mechanisms for emphasizing the most relevant temporal and spatial relationships within the graphs, thereby allowing them to learn how the movements change with time. Being computationally expensive, a sparsity of sensor data can further complicate their operation. An Inflated 3D Convolutional Network (I3D) is a model for classification that utilizes 3D convolutions [53] to capture spatio-temporal features from videos by transforming pretrained 2D CNNs, to handle video frames as a 3D volume. The overfitting problem is the major concern in I3D, i.e., the model may overfit the training data by capturing noise and extraneous information, which can result in weak performance when applied to unseen data. Another network, the Slow–Fast Network [54], is used for HAR by leveraging the advantages of slow and fast processing paths for recognizing a wide range of activities. Slow–Fast Networks involve several complex architectures. This complexity makes them harder to deploy on devices with limited hardware.

A comparative analysis of Human Activity Recognition models based on the latest research is presented in Table 1.

Table 1. A comparative survey of different Human Activity Recognition (HAR) models.

Authors	Dataset Used	Objectives and Proposed Method	Performance
[55]	Weizmann, Ballet and KTH	The author proposed a new method of recognizing human activity in a video sequence using critical poses of the human. silhouettes using Support Vector Machine (SVM)-Neural Network (NN) with Principal Component Analysis (PCA).	Overall accuracy: 96.70%
[56]	Opportunity activity recognition and hand gesture	Deep Convolutional Neural Networks (CNNs) automate feature learning from raw data.	Average Accuracy: 92.11%
[57]	Triaxial gyroscope and accelerometer	An approach to select discriminant features to construct a better generalizable online activity. recognizer was presented for smart phone-based HAR system.	Accuracy: 84.3% Precision: 84.5 and F1 Score: 84.3%

Table 1. *Cont.*

Authors	Dataset Used	Objectives and Proposed Method	Performance
[58]	Weizmann, Ballet and KTH	The AESI, SDP, and modified SDG evidence were combined with motion evidence to. recognize human activity according to a new framework i.e., SVM-KNN with PCA.	Accuracy: 95.83%
[59]	HASC corpus	A Deep Recurrent Neural Network (DRNN) recognizes human activity with high throughput.	Accuracy: 95.42%
[60]	UTKinect Action-3D and CAD-60	Human daily activities can be recognized by tracking skeletal joint movements using a depth sensor.	Accuracy: 97.00%
[61]	Opportunity, Skoda and Actitracker	CNN ensembles have been proposed to resolve the confusion between walking and going upstairs, two highly similar activities. CNN (two variants, CNN-2 and CNN-7) were proposed for recognizing human activities.	Accuracy: 96.11%
[62]	UCI-HAR, PAMAP2 and, WISDM	An end-to-end model using CNN and Gated Recurrent Units (GRUs) is proposed that performs automated feature extraction and activity classification.	Accuracy: 96.20%, 95.27% and 97.21%
[63]	UTKinect Action3D dataset	The joint angle shift method is combined with a Deep Convolutional Neural Network (DCNN) model to classify 3D datasets encompassing spatial–temporal information from RGB-D video image data.	Accuracy: 96.72%
[64]	Conducted real-life experiments of activities.	A muscle–machine interface is constructed using a Bi-Directional Long Short-Term Memory (BiLSTM) network, which decodes multichannel Surface Electromyography (sEMG) signals into flexion and extension angles of the hip and knee joints in the sagittal plane.	Accuracy: 92.8%
[65]	UCF 101 and Kinetics-400 dataset	Firefly Optimization Algorithm is applied to fine-tune the hyper parameters of both the graph-based model and a CNN baseline for comparison.	Accuracy: 88.9%

3. Proposed Methodology

In the proposed methodology, the human body is represented by rectangular patches. Our technique is based on the positions and orientation of these rectangular patches. These rectangular patches are changed over time concerning their actions. Our methodology first extracts the rectangular patches from the video frames over time t. Each video frame has a spatial feature concerning the orientations.

As the rectangular patches are grouped over orientations, a spatial histogram is formed, which can be evaluated over time to see how its shape changes. The human body's location in the given videos is first identified as part of this. A bounding box is then drawn around its silhouette obtained through a background-subtraction approach. Bins of $N \times N$ equal dimensions are then defined within this bounding box. The ratio between the torso, legs and head is considered when forming these spatial bins. The orientations of the rectangular

patches in all spatial bins are used to create a histogram. H_t is used to represent a pose. Firstly, the frames extracted from any video of the datasets as shown in Figure 1. Each video frame is extracted as its feature. In Figure 2, the rectangle histograms are depicted based on spatial orientation.

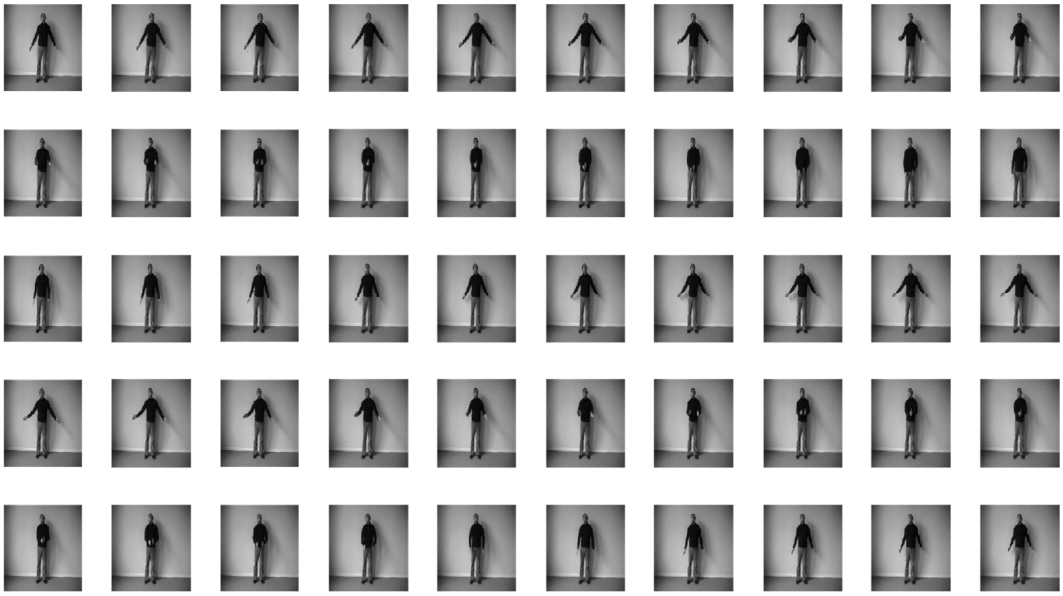


Figure 1. Sequence of frames extracted from the video.

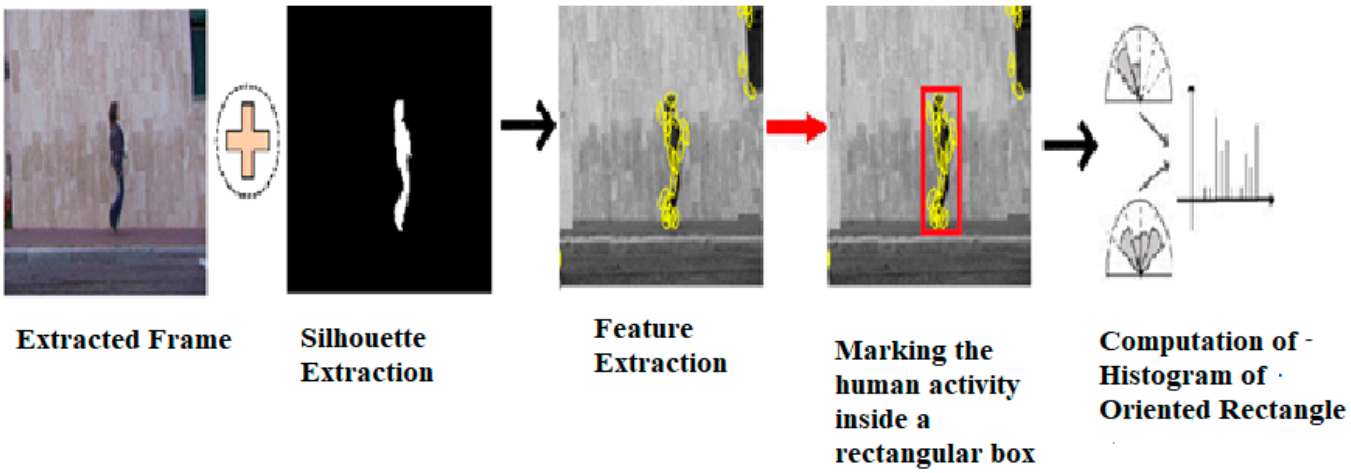


Figure 2. Feature extraction procedure of the proposed approach.

A feature point of the human activity is marked on each extracted frame using background-subtraction technique, as shown in Figure 2. Human activity is marked in the rectangle box, and the grid circumscribing the silhouette is divided into equal parts. A histogram is then computed for each oriented rectangle within that region. A detailed block diagram of the proposed model is shown in Figure 3. First and foremost, the frames of the video sequence are extracted, as shown in Figure 1. Then, the foreground and background are differentiated and a silhouette is extracted. The rectangle is extracted from the silhouette, as mentioned in the next section. Each rectangle is represented by a histogram based on different orientations, as explained in Section 3.2. A 3D CNN model is built for testing and training. Finally, the classification is performed by the KNN classifier.

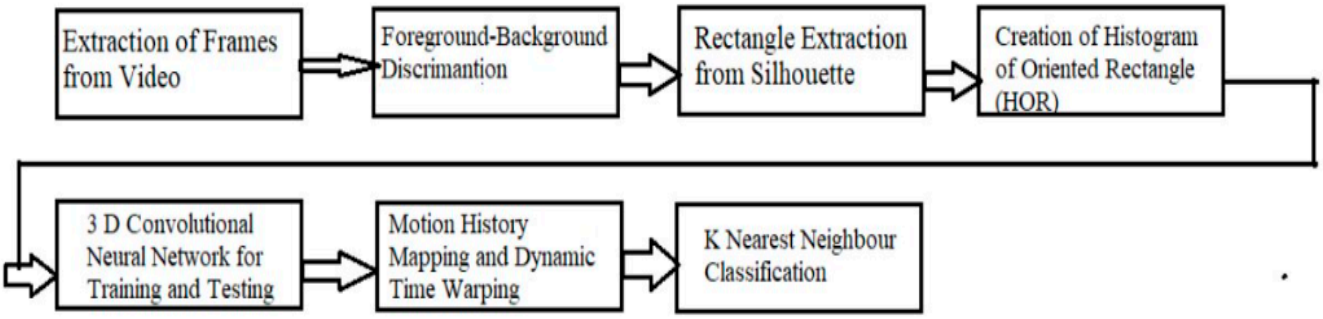


Figure 3. Detailed step-by-step block diagram of the proposed model.

3.1. Rectangle Extraction

Rectangular patches are used to describe human poses. Foreground–background discrimination is usually used to accomplish this. It is the easiest to use background subtraction after forming a reliable background model. We propose a technique for locating moving objects using background-subtraction.

A 3D Convolution Neural Network (CNN) is then used to find rectangular regions over the human silhouette based on the different orientations and scales. It covers a search space of 180° using 12 tilting angles spaced 15° apart. The orientation scales of the rectangular patches are covered at 180°. A rectangular Gaussian filter G_{rect} is used for performing convolutions of the binary silhouette image $I(x, y)$ to create a rectangular region $R(x, y)$.

$$R(x,y) = G_{rect}(x,y) \times I(x,y) \quad (1)$$

A 2 *D* Gaussian $G(x, y)$ patch is applied to G_{rect} to form a zero-padded rectangle.

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \quad (2)$$

There is a greater chance that patches of a particular kind will be found in areas with a higher response to this filter. In the rectangle extraction phase, approximately ∼1000 rectangles are extracted per frame. A histogram can be formed using all of these rectangles by representing each limb with a representative rectangle. In rectangular filters, a representative rectangle covers the silhouette completely. Those candidates with higher responses that are further apart are chosen to meet these constraints. Each frame contains only ∼10 rectangles.

3.2. Histograms of Oriented Rectangles (HORs)—Pose Descriptor

The Histogram of Oriented Rectangles (HOR) can be used to define poses after identifying the rectangular patches on the human body. The orientation of rectangular patches is used to calculate a histogram. Each rectangle is represented by a histogram with twelve circular bins based on 15° orientations. This circular histogram is evaluated on an $N \times N$ grid over the human body to account for spatial information. The best results are obtained with $N = 3$. This grid is created by splitting the silhouette in the Y dimension according to the leg length. A right-to-left and top-to-bottom equalization has been applied to Figure 4. As a result, we provide action space for the arms above the head by allowing some space.

The feature vector is formed by concatenating the HORs of each spatial bin. A coarser grid on the human body and coarser granularity in orientation bins are also examined, which provide more compact features but coarser pose details.

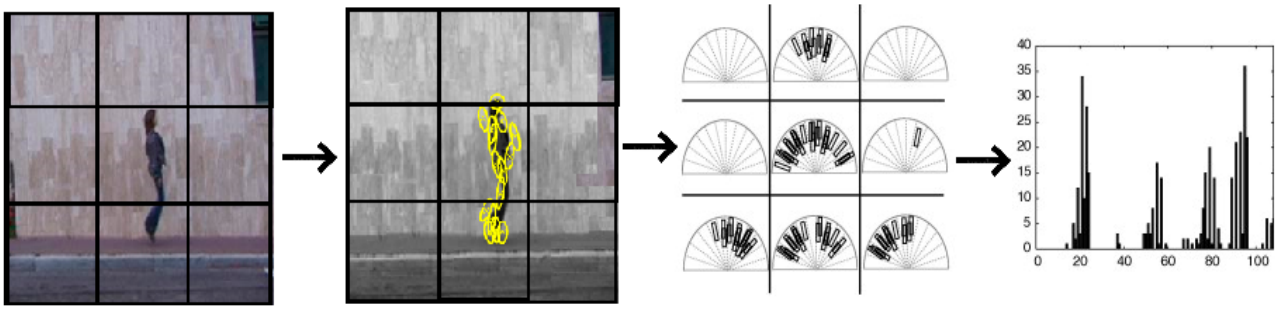


Figure 4. The HORs represent the bounding box around the human action, and feature points are divided into an $N \times N$ grid.

3.3. Capturing Local Dynamics

It is impossible to distinguish two actions based on a single pose in action recognition. A shape-based action descriptor is insufficient since temporal dynamics must also be considered. Rather than using single frames to calculate the HOR, snippets of frames can be used. A HORW is a histogram of rectangles over windows of frames, where every ith the frame has a histogram of rectangles.

$$\mathrm{HORW(i)} = \sum_{\mathrm{k=i-n}}^{\mathrm{i}} \mathrm{HOR(k)} \tag{3}$$

The HOR over this window of frames captures the local dynamics over n frames. According to our experiments, HORW is more helpful in discriminating human actions that are similar in pose domain but differ in speed.

3.4. Train and Test Model

Two-dimensional Convolutional Neural Networks: The convolutional layer of 2D CNNs implements 2D convolution to extract local neighborhood features from the precious layer's feature maps. A sigmoid function and an additive bias are applied to the outcome. The ith layer and jth feature map, u_{ij}^{xy}, describing the unit at the position (x, y), are given by:

$$\mathrm{u_{ij}^{xy}} = \tanh\left(\mathrm{b_{ij}} + \sum_{\mathrm{m}} \sum_{\mathrm{p=0}}^{\mathrm{P_i-1}} \sum_{\mathrm{q=0}}^{\mathrm{Q_i-1}} \mathrm{w_{ijm}^{pq}} \mathrm{v_{(i-1)m}^{(x+p)(y+q)}}\right) \tag{4}$$

A hyperbolic tangent function is defined as $\tanh(.)$, a bias function for this feature map is defined as $\mathrm{b_{ij}}$, a set of feature maps in the $(i-1)$th layer that corresponds to the current feature map is defined as m, $\mathrm{w_{ijm}^{pq}}$ is the position (p, q) of the kernel linked to the mth feature map, and the height and width of the kernel are described by $\mathrm{P_i}$ and $\mathrm{Q_i}$, respectively. In subsampling layers, features are grouped over local neighborhoods, reducing their resolution while making them more insensitive to input distortions. An alternate layer of convolution and subsampling can be used to build CNN architectures. It is usually possible to train the parameters of a CNN either supervised or unsupervised—for example, bias $\mathrm{b_{ij}}$ and kernel weight $\mathrm{w_{ijm}^{pq}}$ [66,67].

Three-dimensional Convolutional Neural Networks: The CNNs compute features from 2D feature maps based on spatial dimensions. It is appropriate to capture the angle information of numerous contiguous frames when analyzing video action detection and recognition problems. The proposed method includes 3D convolutions and orientation changes for computing features from CNNs during the convolution stages. In 3D convolution, several contiguous frames are stacked to form a cube convolved with a 3D kernel. A convolution layer captures motion information by connecting various contiguous frames in

a previous layer. In the ith and jth layers, position (x, y, α) is represented by the equation mentioned below.

$$u_{ij}^{xy} = \tanh\left(b_{ij} + \sum_{m} \sum_{p=0}^{P_i-1} \sum_{q=0}^{Q_i-1} \sum_{r=0}^{R_i-1} w_{ijm}^{pq} v_{(i-1)m}^{(x+p)(y+q)}\right) \quad (5)$$

where R_i is the size of the 3D kernel along with its orientation change. In the previous layer, w_{ijm}^{pq} represents the mth feature map connected to the (p, q, r)th kernel. This network consists of M cliques that cooperate to complete the final task. Cliques are subparts of networks stacked over multiple layers. Cliques extract features from video segments associated with subactions within the overall activity. This method uses a gray scale and orientation change input, with at least m video frames each. To mitigate local body deformations and noise in our proposed model, we apply a max-pooling operator to each 3D convolutional layer. Two fully connected layers are developed next to correlate activity labels with the convolution results of different cliques. Table 2 shown below describes the details of the 3D CNN of the proposed model.

As shown in Figure 5, these components are further defined in the following paragraphs.

Table 2. Detailed architecture of a 3D-CNN designed for HAR.

Layer Name	Stride	Input Dimension	Output Dimension	Filter Size	Parameter Size	Activation Function	Filter/Kernel Size
Conv 3D Layer1	(1,1,1)	(16,64,64,1)	(14,62,62,20)	(3,3,3)	560	ReLu Activation	Convolution layer with 20 filters.
Max Pooling 3D Layer1	(2,2,2)	(14,62,62,20)	(6,30,30,20)		-	-	Max pooling with 3×3 Kernel.
Conv 3D Layer2	(1,1,1)	(6,30,30,20)	(4,28,28,20)	(3,3,3)	5420	ReLu Activation	Second convolution layer with 20 filters.
Max Pooling 3D Layer2	(2,2,2)	(4,28,28,20)	(1,13,13,20)		-	-	Max pooling to reduce the spatial dimensions further.
Flatten Layer	-	(1,13,13,20)	(1,3380)	(3,3,3)	-	-	To flatten the output for the fully connected layers.
Fully Connected Layer1		(1,3380)	512	(3,3,3)	172,352	ReLu Activation	Fully connected layer 1 with 512 neurons.
Fully Connected Layer2		512	128		65,664	ReLu Activation	Fully connected layer 2 with 128 neurons.
Output Layer		128	11		1419		This is the final output layer having 11 neurons for 11 classes.

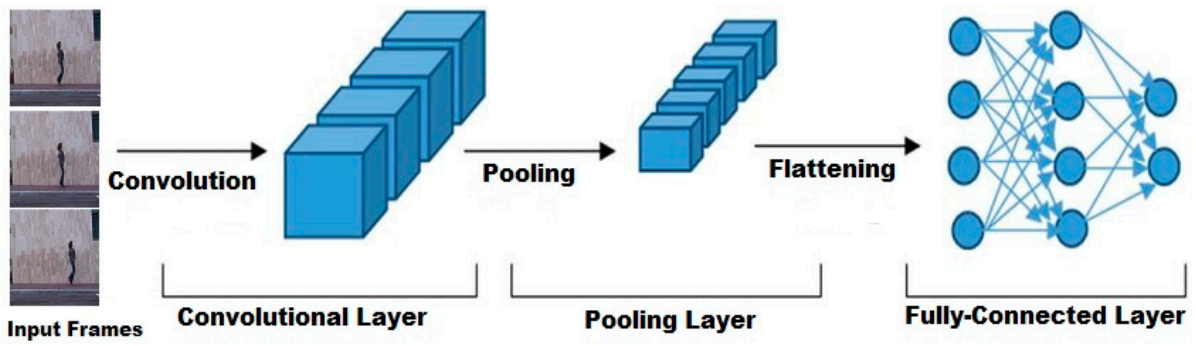


Figure 5. 3D convolution model with its convolutional, pooling and fully connected layer.

- **Max-pooling Operator**

A max-pooling operation follows every 3D convolution. It is possible to determine the invariance of shift and deformation of a system using the proposed method [68,69]. The max-pooling process is used to subsample a set of feature maps, resulting in a similar number of feature maps but with reduced spatial resolution. As an example, if a 2×2 max-pooling operator has been applied to $a_{1\times}a_2$ feature map, then the max values in each 2×2 non-overlapping region of a 2×2 feature map should be accumulated to form a new feature map of size $a_1/2 \times a_2/2$.

- **Full Connection Layer**

The proposed model contains two full connection layers. Feature maps from different M model cliques are first concatenated into an extended feature vector. In addition, each vector unit is connected to the neurons in the first layer of full connections. Each output neuron signifies the probability of an action hypothesis, and as a result, each output neuron K represents the number of groups of actions. The softmax function is applied to the output labels to normalize their probability.

$$\sigma(z_i) = \frac{\exp(z_i)}{\sum_{k=1}^{K} \exp(z_k)}, \tag{6}$$

In this formula, z_i is the value of the ith output neuron calculated by multiplying its values with its weights, and $\sigma(z_i)$ is its probability. Note that $\sum_{k=1}^{K} \exp(z_k) = 1$. Each video frame contains a grey and depth channel. The grey and depth images are obtained from the raw data. The first layer of the convolutional kernel duplicates the channels. The convolved feature maps are the same size since the convolution results are combined. It is also possible to apply the proposed model to multiple-channel video frames.

3.5. Oriented Rectangle Histograms: Recognizing Activities

To evaluate the proposed pose descriptor's performance for every video frame in the action classification, we incorporate the four matching methods after estimating pose descriptors for every video frame. To simplify the process, we consider single frames (or an extracted number of frames) as the matching procedure criteria, disregarding the sequence's dynamics. In the test instance, we label each frame with its database is classified based on its nearest neighbor.

The nearest neighbor classification employs voting techniques throughout the entire sequence. A jogging sequence from the KTH database is classified based on its nearest neighbor, as mentioned in Figure 6.

Histogram distance is used to compute the X^2 distance between frames [70]. Frames with histograms H_i are labelled with the class of the frame, with histograms H_j having the smallest distance X^2 among them.

$$X^2(H_i, H_j) = \frac{1}{2}\sum_{n} \frac{(H_i(n) - H_j(n))^2}{H_i(n) + H_j(n)} \tag{7}$$

Due to variations in binning the rectangles when the bounding box center of the human silhouette changes, X^2 and L_2 distance functions are very susceptible to noise. For these, we consider the distances among the bins at the expense of the longer computation time, and we can use the Earth Mover's Distance [71] or Diffusion Distance [72], which have been shown to be most proficient for histogram assessment in the presence of shifts.

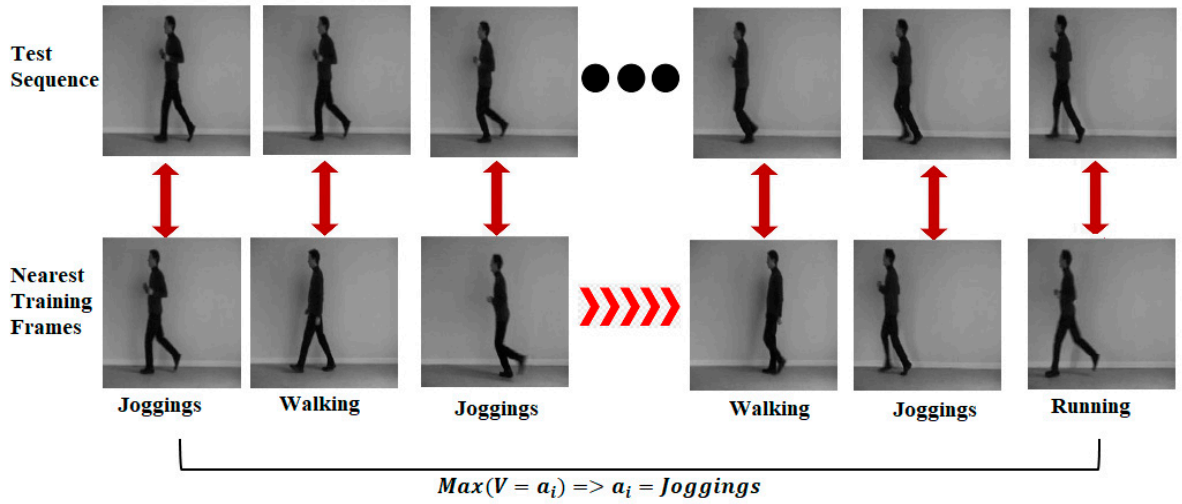


Figure 6. Classification of KTH jogging sequences by nearest neighbors.

Global Histogramming: A global histogram method [73] is proposed using an approach similar to a Motion Energy Image (MEI). Motion energy mapping is a technique in which the dynamic energy of the movement is visualized and analyzed using a spatial map. A motion energy template provides a high-level representation of the video. All spatial histograms from the oriented rectangles in the sequence are summed up to represent the entire video compactly. A global histogram is formed based on each d dimension by summing all the histograms in H_{global} and forming a global histogram based on the sum.

$$H_{glabal}(d) = \sum_{t} H_t(d) \tag{8}$$

There is a link between each test instance's H_{global} and the training instances X^2, fomed using distances and a description of the nearby contests.

It is most efficient to use a global histogramming method with long sequences that captures the associated pose information with each pass. A histogram that tracks the accumulated results of a series of actions is more accurate when the sequences are long enough, such as in the case when each action is repeated multiple times. In the case of short sequences and a different number of cycles, the accumulated histogram may not deliver a good picture of the action's structure.

K-Nearest Neighbor Technique Classification: The K-Nearest Neighbor (KNN) is a simple and easy-to-understand method for learning. Meanwhile, KNN has the advantage of being an effective deployment and implementation technique. Additionally, KNN has the advantage of working accurately with multiclasses, achieving better accuracy. Furthermore, it is a simple technique, which earned it a reputation in Machine Learning (ML). Thus, it is easy to implement, in addition to the advantages mentioned earlier. According to the classification approach, the closest training points for the human pose descriptor are used in the classification process. KNNs learn by analogy by comparing test cases with similar-looking training data [74]. Every row represents an n attribute (feature) in the n dimensional space described by the training data. In the first step, KNN searches the dataset with the most similar pattern space to the test dataset. Therefore, this training dataset is considered the K-Nearest Neighbor of the test dataset. Distance metrics are used to compare training and test data. In an n dimensional space, a KNN can also use

the Euclidian distance to find the distance between two points. The Euclidian distance is calculated using Equation (9).

$$\text{dist}(X_1, X_2) = \sqrt{\sum_{i=1}^{n} (X_{1i} - X_{2i})^2} \tag{9}$$

X_1 and X_2 are the two points $X_1 = (X_{11}, X_{12}, \ldots, X_{1n})$, and $X_2 = (X_{21}, X_{22}, \ldots, X_{2n})$. The two-level classification system of the proposed model is based on this observation. According to the average velocities of the training classifications, each action in action set A = $\{a_1 \ldots a_n\}$ is fitted with a Gaussian function. A subsequent probability is calculated for each action $a_i \in A$ over these Gaussians based on the test instance. If the subsequent probability a_i of each action exceeds the threshold, we add it to the probable set of actions for a sequence $t = 0.1$. After preprocessing, we estimate only the $a_k \in S$ output for actions in the second level. KNN classifiers that have the maximum response are selected to be used for classification.

Dynamic Time Warping (DTW): Actions are sequences of poses using the proposed representation. Each pose is represented as a distribution of rectangles. A 3D representation of an action is created by concatenating 2D feature vectors per pose. There is a wide variation in the time and speed of action sequences performed by different humans. Dynamic Time Warping (DTW) is the most popular method for handling similarities among time series. A similar approach to compare three-dimensional representations of action sequences is taken by the authors [75]. It compares two different 2D action representations located at the location of the exact rectangle with the help of DTW. We measure fluctuations at the exact rectangle location over time using a specific orientation and size of rectangles. The voxel grid of some body regions does not change throughout an action. In this way, some boxes in the rectangle histogram rarely change data series. To measure changes in rectangles with different sizes and orientations over time, we measure the variance at each rectangle's location. A time warp is used in this model to align samples of consistent points. Using the specified two time series, $x_1 \ldots x_n$ and $y_1 \ldots y_m$, the distance D(i, j) is calculated, as shown in Equation (10).

$$D(i,j) = \left\{ \begin{array}{c} D(i,j-1) \\ D(i-1,j) \\ D(i-1,j-1) \end{array} \right\} + d\left(x_i, y_j\right) \tag{10}$$

where $d\left(x_i, y_j\right)$ is the local distance function as per the application. The X^2 distance function in the proposed implementation is $d\left(x_i, y_j\right)$, as given in Equation (7). The histograms are shown in Figure 7 with DTW along each dimension. Each 2D series of the histogram react angles in the test video has a DTW distance of D(X(d), Y(d)) compared to its corresponding 2D series in the training video. The DTW distance D_{global} between two videos is computed by adding both dimensions' distances together. Training video instances are labelled with the smallest D_{global} label.

$$D_{global}(X, Y) = \sum_{d=1}^{M} D(X(d), Y(d)) \tag{11}$$

Using the total number of rectangles in the histograms, we exclude the top k distances to reduce the effects of shifted rectangles. Accordingly, we choose $k = \left\lfloor \#num_{rectangles}/2 \right\rfloor$ with $\#num_{rectangles}$ as the total rectangles in the spatial grid according to the feature vector size.

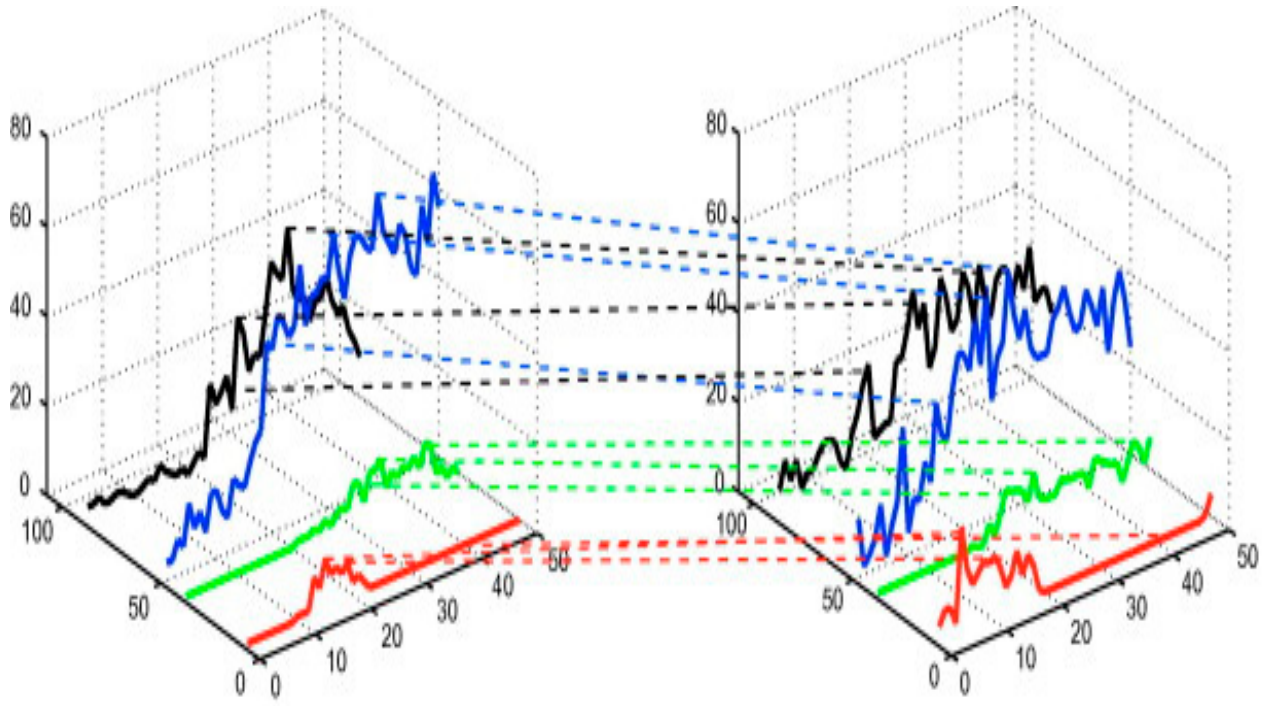


Figure 7. An analysis of 2D histograms using dynamic time warping (DTW).

4. Experiments and Results Analysis

Four publicly available datasets are used to evaluate the Human Action Recognition framework: KTH [76], Weizmann [75] UT-Tower [77] and YouTube. The UT-Tower and Weizmann datasets contain silhouette sequences, whereas the YouTube and KTH datasets do not. The query segment was taken from a single action frame, and all other frames were used as training segments (except those from the same video sequence). The query segment has been categorized based on its action class. Splitting the datasets is crucial for evaluating the model's performance. We applied five-fold cross-validation, since it ensures that the model is evaluated on different data subsets, reducing the bias caused by a single random split. Also, K-Fold splitting provides a robust evaluation by splitting the datasets into K datasets and testing the model on each while training on the remaining data. It also reduces the likelihood of overfitting and offers a very reliable estimate of the performance of the model across different subsets of data.

4.1. Datasets Used

KTH Dataset: KTH actions include six human actions: walking, running, jogging, hand waving, clapping, and boxing. It is one of the most commonly used datasets. It is a dataset in which 25 actors perform each action, and each actor's setting is adjusted systematically per action. S_1 Indicates an outdoor setting, S_2 indicates an outdoor setting with scale variation, S_3 indicates an outdoor setting with different clothing, and S_4 indicates an indoor setting, as shown in Figure 8. A total of 2391 sequences are currently stored in the database. Each sequence was captured with a static camera at a 25 fps frame rate over a homogeneous background. The sequences are downsampled to 160 $\times$ 120 pixels with an average duration of four seconds. The techniques are tested with these variations by identifying actions regardless of the actors' background, appearance or scale.

Weizmann Dataset: There are 2513 images of human activity in the Weizmann dataset. Nine actors represent five types of human behavior. An online dataset called the Weizmann dataset originally contained 81 low-resolution videos of nine subjects performing nine different actions (viz. running, jumping in place, bending, waving with one hand, jumping

jacks, jumping sideways, jumping forward, walking, and waving with two hands). A 10th action was then added: jumping on one leg. Figure 9 illustrates each of these actions with examples.

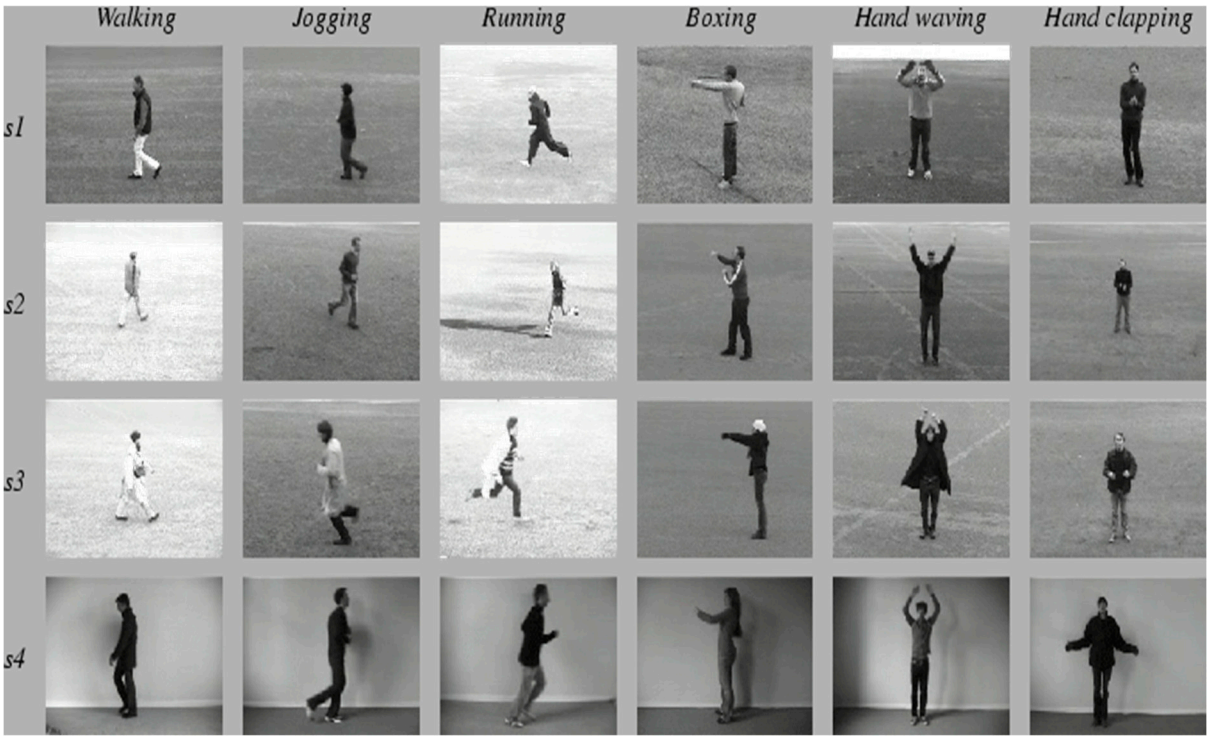


Figure 8. KTH dataset with its six types of human actions.



Figure 9. Weizmann dataset with its ten different human actions.

UT-Tower Dataset: As part of the Semantic Description of Human Activities (SDHA) competition at ICPR 2010, the UT-Tower action dataset is used. With a pixel resolution of 1080 and a frame rate of 59 frames per second, the dataset contains 108 low-resolution video sequences. During the contest, video sequences are categorized according to nine types of human actions. Foreground masks, bounding boxes, and ground truth action labels accompany the video sequences. There are nine categories of human actions shown: pointing, standing, digging, walking, carrying, running, wave 1, wave 2, and jumping. Each of the nine actions was recorded in 12 video sequences. For each action category, six

individuals performed them twice. The grass is used for carrying, running, wave1, wave2, jumping, and pointing, while concrete is used for standing, digging, and walking videos.

YouTube Dataset: The dataset based on YouTube videos is called the YouTube dataset. The 11 action classes in the dataset are bicycle riding, diving, basketball shooting, golf swinging, soccer juggling, horses, trampoline jumping, tennis swinging, walking with a dog, and volleyball spiking. The sample images of the YouTube dataset are presented in Figure 10. There are numerous variations in camera motion, cluttered background, acquisition viewpoint, etc.

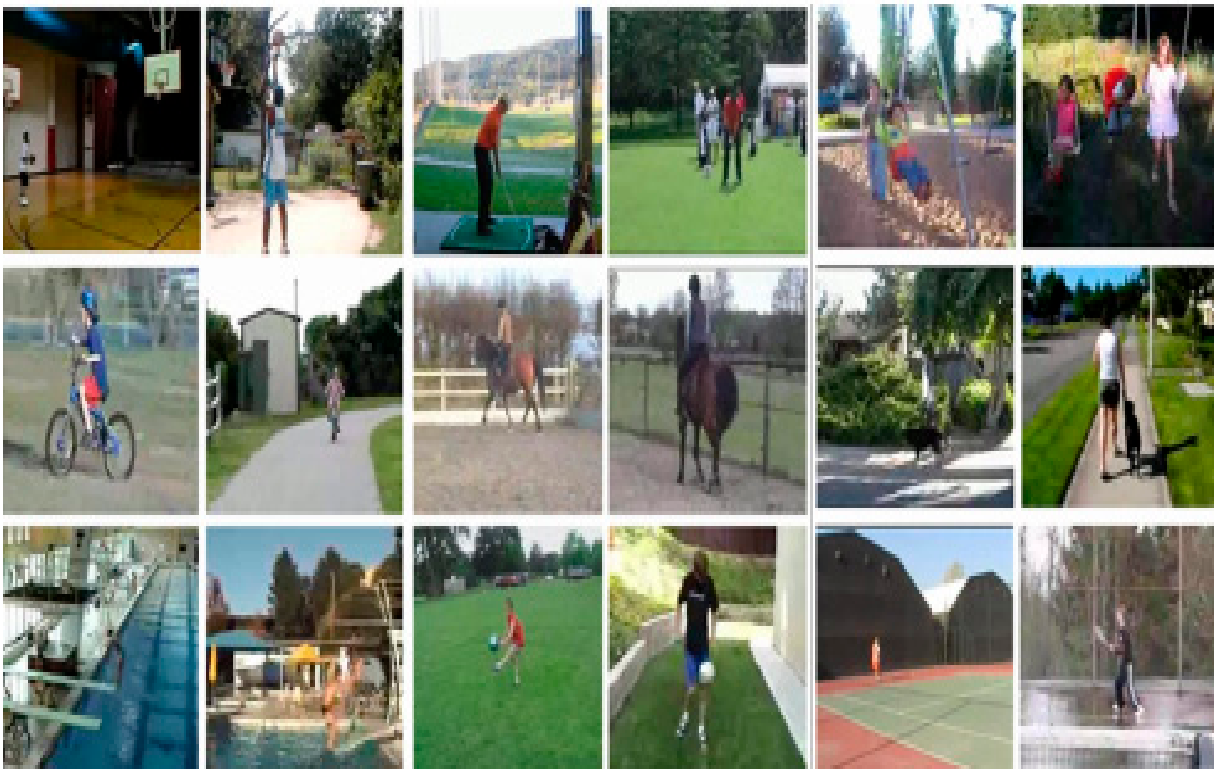


Figure 10. YouTube dataset with its different human actions.

4.2. Performance Evaluation Metrics

The proposed model was evaluated using evaluation metrics, such as Classification Correction Rate (CCR) and training loss. True Positives (TPs) and True Negatives (TNs) are recognized correctly, and False Positives (FPs) and False Negatives (FNs) are recognized incorrectly. The model is specifically proposed for accuracy sensitive applications.

Classification Correction Rate (CCR): The Correction Classification Rate (CCR) of a proposed model is calculated by adding up the True-Positive and True-Negative samples and dividing them by the sum of all samples—viz. TP, FP, TN, and FN, as mentioned below in Equation (12). The CCR is the percentage of correctly classified query segments in the dataset.

$$\mathrm{CCR}(\%) = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{TN} + \mathrm{FN}} \times 100 \tag{12}$$

Recall (R): The model's sensitivity is measured with this metric. As a proportion of the total number of actions, the recall represents how many anomalous actions were detected, as mentioned in Equation (13).

$$\mathrm{Recall(R)} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{13}$$

Precision (P): In terms of precision, it refers to the number of original actions detected among those detected. Measuring false alarms directly correlates with precision metrics. As false alarms are detected over the correct detection of anomalies, False Positives (FPs) are measured as mentioned in Equation (14).

$$\text{Precision(P)} = \frac{\text{TP}}{(\text{TP} + \text{FP})} \tag{14}$$

The F1 score is an important metric that combines both precision and recall, resulting in a a single score, as mentioned below in Equation (15).

$$\text{F1 score} = (2 \times \text{Precision} \times \text{Recall})/(\text{Precision} + \text{Recall}) \tag{15}$$

A confusion matrix is used to evaluate the performance of the model using TPs, FPs, FNs and TNs, as shown below. The diagonal elements are correctly predicted instances and off-diagonal elements are misclassification values.

TP	FN
FP	TN

Since this model is designed for improving accuracy, all the CCR values are compared.

4.3. Result Analysis and Discussion

This section describes our experimental results and examines the four publicly available human activity datasets for different contexts. A CNN comprises a convolution layer with partial weight sharing, employing a combination of 20 filters and a pool size of three. The reason for taking a pooling size of three is that a larger window, such as 4×4 or 5×5, would result in too much downsampling, thereby losing the finer details. However, taking a smaller window, such as 2×2, would preserve the information spatially, thereby reducing the effectiveness of dimensionality reduction, and it would also make the model less computationally efficient. The size of the filter plays an important role in founding the key features. The reason for using 20 filters in our model is that it allows the model to learn a wider range of features which is crucial for recognizing human activities, thereby leading to greater accuracy. The choice of taking 20 filters is also due to maintaining a balance between the model's ability to learn and the computational efficiency. A top layer with a softmax is added to the two fully connected layers for generating state posterior probabilities. Human activity is classified using KNN as a label predictor.

A system equipped with a 64-bit processor with DDR4 RAM up to 2400 MHz and 64 GB and an Intel Core i7 was used to conduct this research. Since the work as conducted on i7, the training time for the model is 6 h. However, this training time could be significantly reduced by using GPU. The initial learning rate taken was 0.001. An adaptive learning rate optimizer was used as it adapts the learning rate. The learning rate was exponentially decayed as the model trained to allow for even finer adjustments while reaching the end of the training process. All the CNNs are trained for 50 epochs using cross-entropy loss and optimizing the network. These parameters were selected for every CNN trained to improve its convergence. The proposed method improved CCR when compared to the KTH, Weizmann, UT-Tower and YouTube datasets, as shown in Figure 11. The Weizmann dataset's CCR is 98.88%, which higher than that of the KTH, UT-Tower and YouTube datasets. The accuracy increased rapidly just after five epochs for all four datasets. The Weizmann and KTH datasets have low training loss and higher classification accuracy for Human Action Recognition.

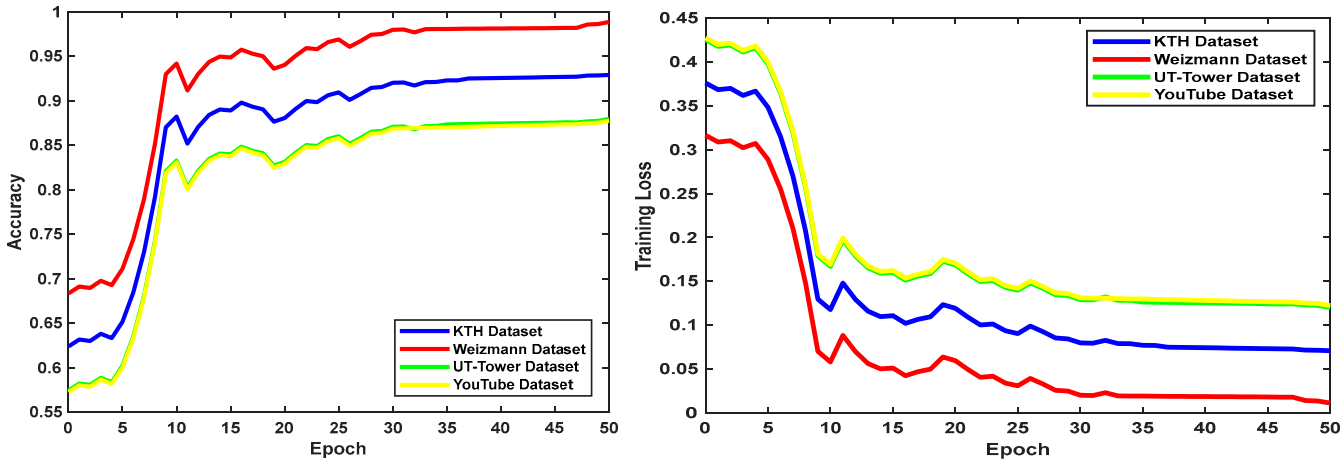


Figure 11. Comparative analysis of CCR and training loss of the proposed model for all four datasets.

When the calculated CCR is higher, human activities are classified more correctly. Classifying an entire sequence of frames containing an action is usually more helpful than just one in isolation. First, frames from each video sequence are obtained to classify the entire sequence since they provide time-localized action information. Figure 12 shows the CCR performance analysis based on the KTH dataset's action. The running and hand-waving actions show a higher rate of CCR (%), but walking and handclapping show a similar performance. In the KTH dataset, boxing has a lower percentage recognition rate of 79% because boxing activity involves swift hand movements. The overall performance of the proposed model is highly acceptable because the lowest recognition rate is 79%, and the highest is 97.4% for the KTH dataset.

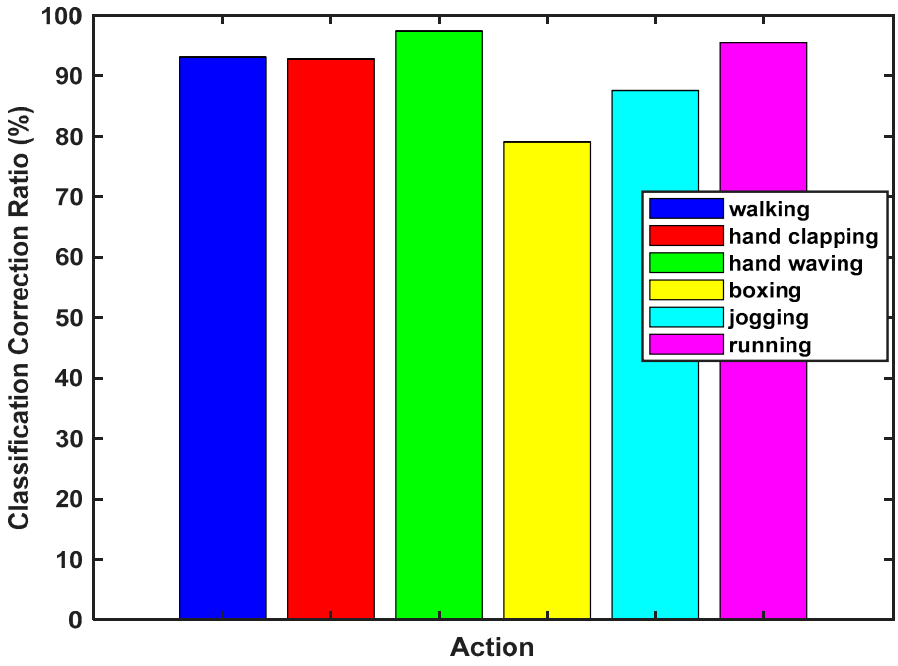


Figure 12. Performance evaluation of different activities in terms of CCR (%) on KTH dataset.

For a quantitative comparison, we have also presented a comparative analysis of the KTH dataset, shown in Table 3. The proposed model's performance is compared to NN classifiers [78], SLA classifiers [79] Wong et al. [80] and Dollar et al. The proposed technique shows a higher classification rate as compared to the existing techniques for the

KTH dataset. The proposed technique is remarkable because it outperforms all existing techniques with a CCR of 92.91%. Figure 13 illustrates Table 3 graphically.

Table 3. Comparative analysis of proposed model with state-of-the-art models on KTH dataset.

Authors	Proposed Method	NN Classifier	SLA Classifier	Wong et al.	Dollar et al.
CCR (%)	92.91	89.55	90.84	81	81.2

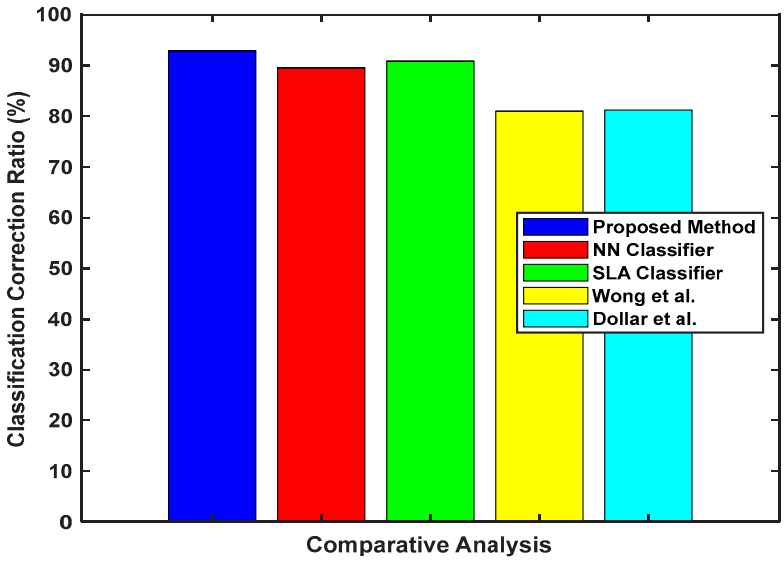


Figure 13. Comparative performance analysis with different state-of-the-art models based on CCR (%) on KTH dataset.

Figure 14 shows Weizmann dataset's calculated CCR (%) that correctly classified the action. For bend, jack, sjump, side, walk, wave1 and wave2 human activities, the proposed technique achieved a CCR of 99.6%, 99.5%, 99.5%, 98.9%, 99.0%, and 97.4%. This suggests that jumps, runs, and skips are harder to distinguish from other actions, such as jacks, bends, sjumps, walks, wave1, side and waves. The side, bend, jack, and sjump actions have a higher recognition rate in the Weizmann dataset. Similarly, skip and run show the lowest recognition rate because of the video speed and orientation mismatch.

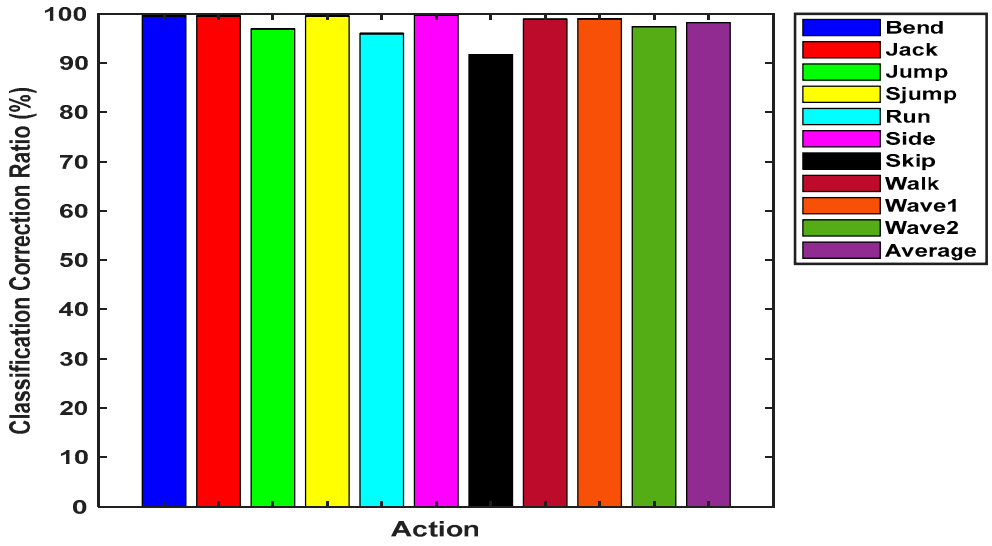


Figure 14. Comparative analysis of different activities based on CCR (%) on Weizmann dataset.

Compared with some current state-of-the-art techniques, the proposed technique performs better. The proposed technique is compared with the existing techniques of NN, SLA, Gorelick et al. and Ali et al. [81] The proposed technique shows a CCR performance of 98.88%, which is higher than that of the existing techniques. Figure 15 shows a graphical representation of Table 4.

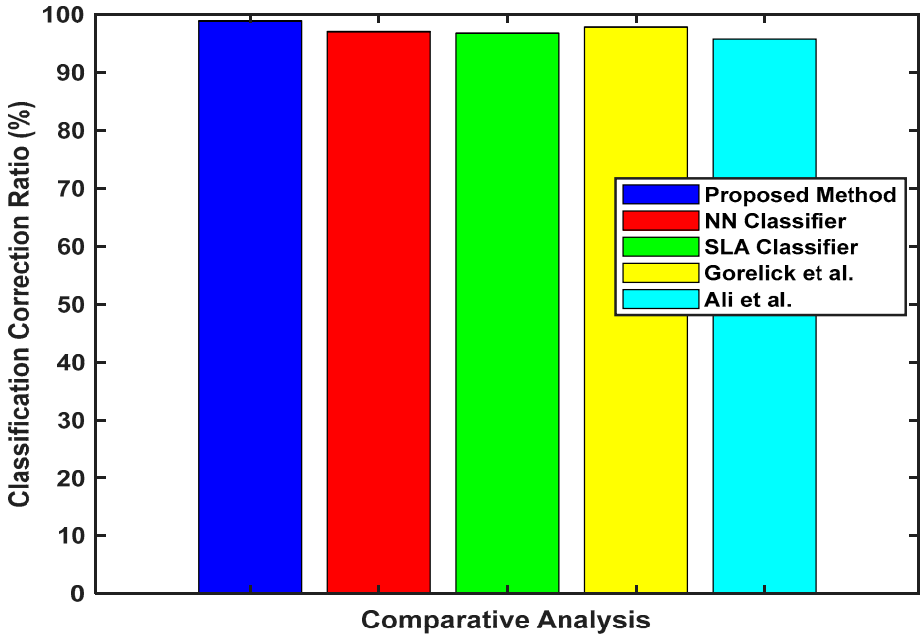


Figure 15. Comparative performance analysis with state-of-the-art models in terms of CCR (%) on Weizmann dataset.

Table 4. Comparison with state-of-the-art models on Weizmann Dataset.

Authors	Proposed Method	NN Classifier	SLA Classifier	Gorelick et al.	Ali et al.
CCR (%)	98.88	97.05	96.74	97.83	95.75

Figure 16 shows the calculated CCR (%) for the UT-Tower dataset, correctly classifying the action. The proposed technique achieved a CCR (%) of 76.30, 96.80, 98.50, 99.97, 99.75, 89.60, 99.71, and 99.57 for pointing, standing, digging, walking, carrying, running, wave1, wave2, and jumping. This indicates that some actions, such as jumping, pointing and running, are more difficult to distinguish from others as they show the lowest recognition rate in the UT-Tower dataset.

The proposed model is quantitatively compared on the UT-Tower dataset with the existing models in the literature, as shown in Table 5. The proposed technique's performance is compared with NN classifiers, Liu et al., Ikizler et al. [82], Le et al. [83], Wang et al. [84], Noguchi et al. [85] and techniques based on the Classification Correction Rate (CCR) (%). The proposed technique's performance is remarkable because it outperforms all existing techniques, with a CCR of 87.97%. Figure 17 shows a graphical representation of Table 5.

Table 5. Comparative analysis with other models on UT-Tower Dataset.

Authors	Proposed Method	NN Classifier	Liu et al.	Ikizler et al.	Le et al.	Wang et al.	Noguchi et al.
CCR (%)	87.97	78.5	71.2	75.2	75.8	84.2	80.4

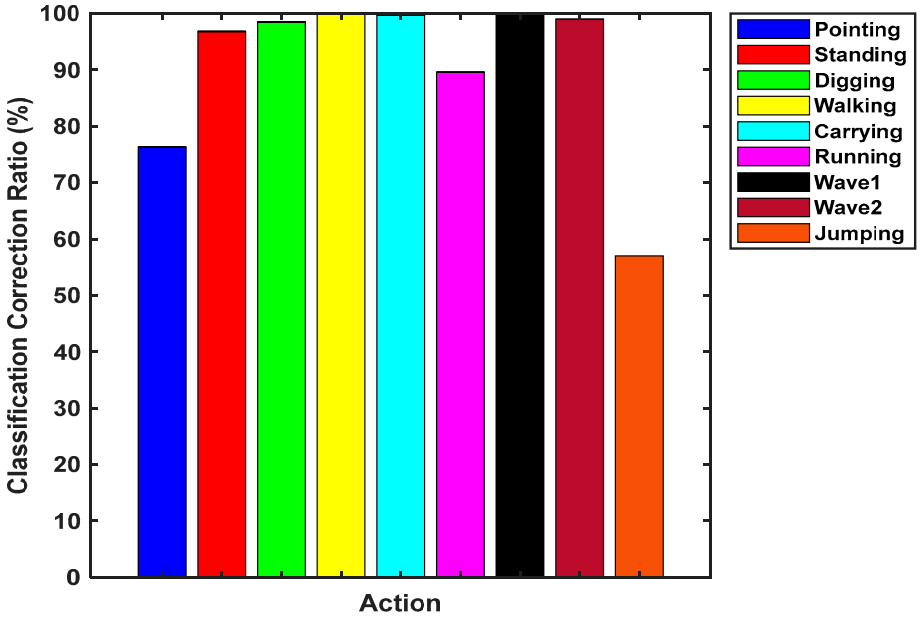


Figure 16. Comparative analysis of different activities in terms of CCR (%) on UT-Tower dataset.

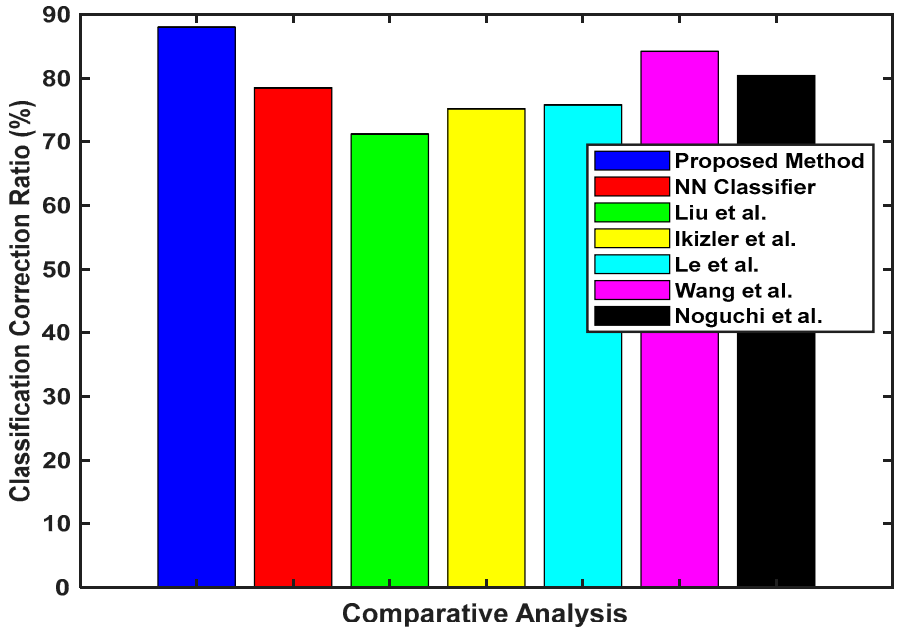


Figure 17. Comparative performance analysis with state-of-the-art models in terms of CCR (%) on UT-Tower dataset.

Figure 18 shows the CCR (%) for the YouTube dataset that correctly classified the activity. The proposed technique achieved a CCR (%) of 77.70, 78.89, 94.80, 77.71, 88.55, 79.67, 89.57, 86.96, 95.55, 79.17, and 89,77 for basketball-shooting, biking/cycling, diving, golf swinging, horse riding, soccer juggling, swinging, tennis swinging, trampoline jumping, volleyball spiking, and walking with a dog. The YouTube dataset has complex and various activities that are difficult to recognize, but the proposed model shows more than a 75% performance improvement ratio. The driving, trampoline jumping, and walking with dog activities have a CCR rate of more than 90%.

Table 6 and Figure 19 show a comparison between the proposed and existing state-of-the-art models on the YouTube dataset. Our proposed model outperforms other state-of-the-art techniques like NN classifiers, Liu et al. [78]. ,Ikizler et al. [82], Le et al. [83], Wang et al. [84], and Noguchi et al. [85].

Table 6. Comparative analysis with other models on YouTube dataset.

Authors	Proposed Method	NN Classifier	Liu et al.	Ikizler et al.	Le et al.	Wang et al.	Noguchi et al.
CCR (%)	87.77	78.5	71.2	75.2	75.8	84.2	80.4

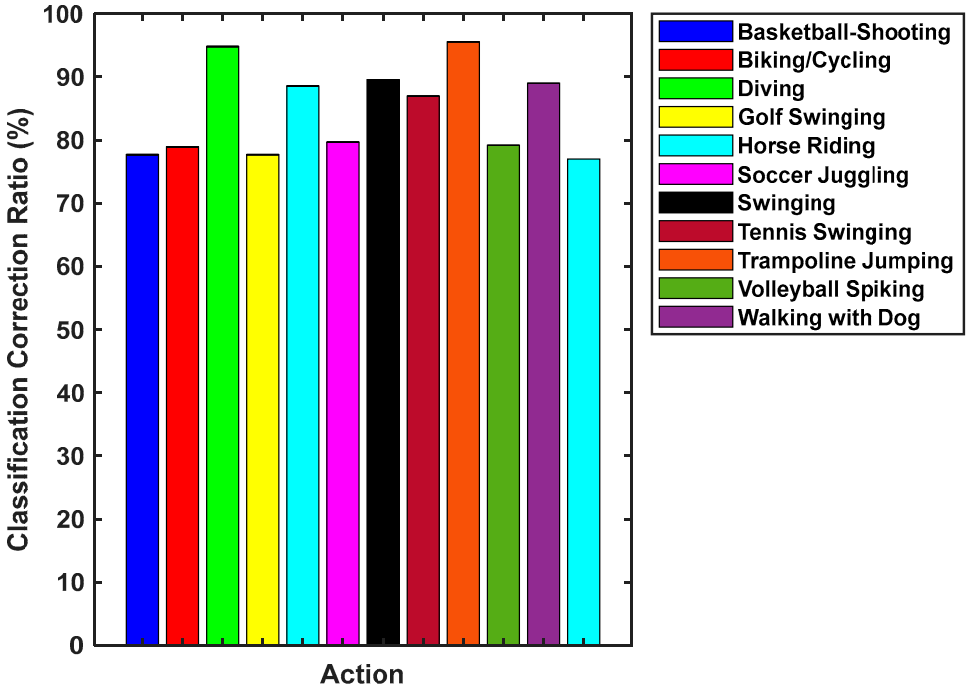


Figure 18. Performance analysis of different activities in terms of CCR (%) on YouTube dataset.

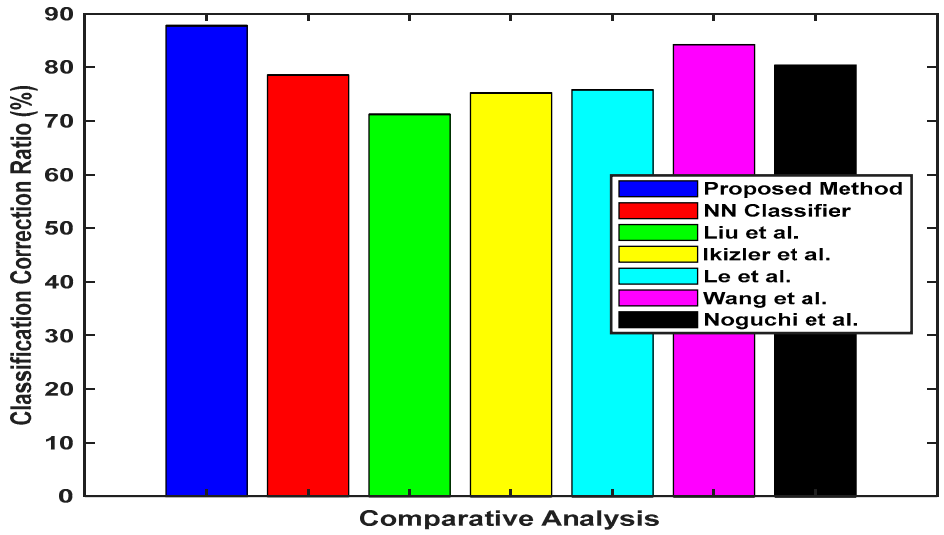


Figure 19. Comparative performance analysis with the state-of-the-art models in terms of CCR (%) on YouTube dataset.

Based on the results on the Weizmann KTH, UT-Tower and YouTube datasets, Figure 20 shows the precision and recall curve for the proposed model. The graphical representation of the precision versus recall represents that the proposed model shows higher performance for the Weizmann dataset and the lowest performance for YouTube dataset. The KTH dataset also shows a higher precision performance versus recall for the proposed approach than the UT-Tower dataset. The combined results are shown in Table 7. This clearly shows that our proposed method significantly outperforms all other methods taken for comparison on the abovementioned datasets.

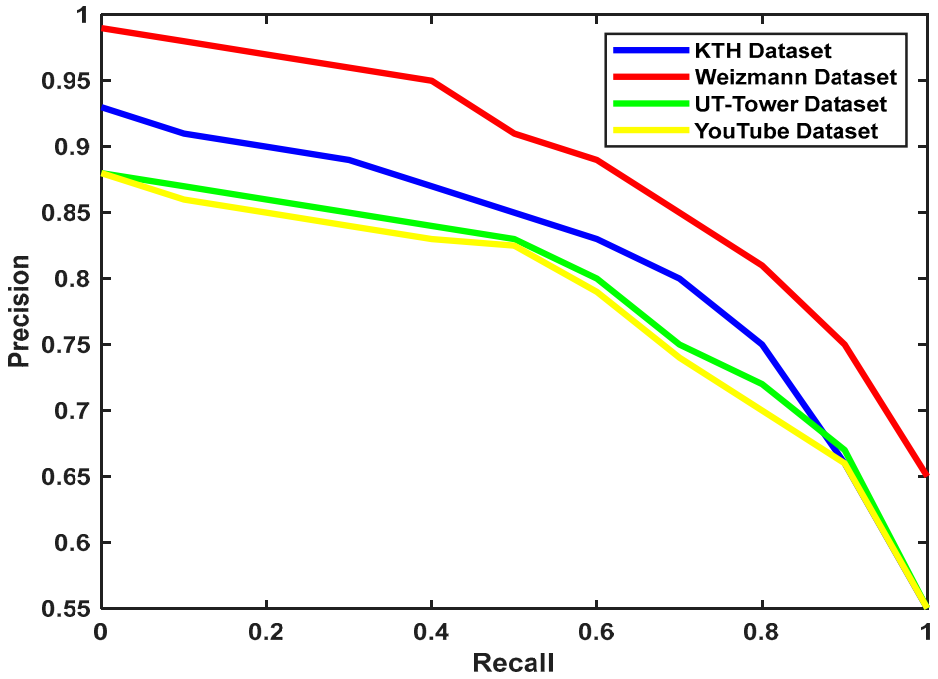


Figure 20. Performance analysis of the proposed model on four datasets in terms of precision and recall.

Table 7. Comparative analysis of our proposed method with state-of-the-art methods.

Authors	Dataset Used for Comparison	Classification Correction Rate (CCR %)
Proposed Method	**KTH Dataset** **Weizmann Dataset** **UT-Tower Dataset** **YouTube Dataset**	**92.91** **98.88** **87.97** **87.77**
NN Classifier	KTH Dataset Weizmann Dataset UT-Tower Dataset YouTube Dataset	89.55 97.05 78.50 78.50
SLA Classifier	KTH Dataset Weizmann Dataset	90.84 96.74
Wong et al.	KTH Dataset	81.00
Dollar et al.	KTH Dataset	81.20
Gorelick et al.	Weizmann Dataset	97.83
Ali et al.	Weizmann Dataset	95.75
Liu et al.	UT-Tower Dataset YouTube Dataset	71.20 71.20
Ikizler et al.	UT-Tower Dataset YouTube Dataset	75.20 75.20
Le et al.	UT-Tower Dataset YouTube Dataset	75.80 75.80
Wang et al.	UT-Tower Dataset YouTube Dataset	84.20 84.20
Noguchi et al.	UT-Tower Dataset YouTube Dataset	80.40 80.40

5. Conclusions and Future Work

The increasing number of surveillance cameras installed in public and private locations has made the computer-based automatic and intelligent video analysis of video sequences increasingly important. This paper proposes a novel approach for Human Action Recognition using a Motion History Mapping and Orientation-based CNN. In the proposed work, the human body is signified by rectangular patches, and this technique is based on the positions and orientation of the rectangular patches. These rectangular patches are changed over time concerning their actions. To evaluate our model's performance, we have conducted extensive experiments on four challenging, widely used and publicly available datasets: KTH, Weizmann, YouTube, and UT-Tower. The obtained experimental results are compared with existing state-of-the-art models. The performance of the proposed model was evaluated in terms of the Classification Correction Rate (CCR). According to the results, our proposed model offers significantly superior and accurate performance when compared to the other models for extracting human actions, i.e., 92.91 (% CCR) for the KTH dataset, 98.88 (% CCR) for the Weizmann dataset, 87.97 (% CCR) for the UT-Tower dataset and 87.77 (% CCR) for the YouTube dataset. The proposed method outperforms the existing methods and is also computationally efficient, thus making our model an important contribution in the field of Human Activity Recognition.

In future work, the proposed model can be combined with the transfer learning approach, where we can pre-train our model on the curated dataset and fine-tune it on a smaller, in-the-wild dataset. This would help the model to leverage the knowledge learnt in the curated dataset and adapt it to new and complex scenarios of the real-world, in-the-wild dataset. This is because the transfer learning approach would allow the model to handle a variety of noise patterns. Also, the proposed work can be extended by using a multi-view fusion approach. This technique can significantly reduce the memory and computational requirements while still capturing wide spatial information. The limitation of our model is that the 3D CNNs lack an attention mechanism. The attention-based deep learning technique allows the network to emphasize important features while suppressing irrelevant ones. Attention-based deep learning and transformer-based mechanisms could extend this work and further enhance performance.

Author Contributions: I.A. contributed to the conception, design, analysis, interpretation of results, running codes and drafting of the manuscript. M.G. also contributed to result analysis, supervision and investigation. All authors have read and agreed to the published version of the manuscript.

Funding: No funding was received for this manuscript.

Institutional Review Board Statement: This research does not contain any studies with human participants or animals performed by any authors.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare that they have no conflicts of interest.

References

1. Zhu, A.; Wu, Q.; Cui, R.; Wang, T.; Hang, W.; Hua, G.; Snoussi, H. Exploring a rich spatial–temporal dependent relational model for skeleton-based action recognition by bidirectional LSTM-CNN. *Neurocomputing* **2020**, *414*, 90–100. [CrossRef]
2. Zhao, B.; Gong, M.; Li, X. Hierarchical multimodal transformer to summarize videos. *Neurocomputing* **2022**, *468*, 360–369. [CrossRef]
3. Chang, Z.; Ban, X.; Shen, Q.; Guo, J. Research on Three-dimensional Motion History Image Model and Extreme Learning Machine for Human Body Movement Trajectory Recognition. *Math. Probl. Eng.* **2015**, *2015*, 528190. [CrossRef]

4. Kanazawa, A.; Zhang, J.Y.; Felsen, P.; Malik, J. Learning 3D Human Dynamics From Video. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 5607–5616.
5. Çalışkan, A. Detecting human activity types from 3D posture data using deep learning models. *Biomed. Signal Process. Control.* **2023**, *81*, 104479. [CrossRef]
6. Luo, J.; Wang, W.; Qi, H. Spatio-temporal feature extraction and representation for RGB-D human action recognition. *Pattern Recognit. Lett.* **2014**, *50*, 139–148. [CrossRef]
7. Hu, G.; Cui, B.; Yu, S. Joint Learning in the Spatio-Temporal and Frequency Domains for Skeleton-Based Action Recognition. *IEEE Trans. Multimed.* **2020**, *22*, 2207–2220. [CrossRef]
8. Wang, B.; Ye, M.; Li, X.; Zhao, F.; Ding, J. Abnormal crowd behavior detection using high-frequency and spatio-temporal features. *Mach. Vis. Appl.* **2012**, *23*, 501–511. [CrossRef]
9. Péteri, R.; Chetverikov, D. Dynamic Texture Recognition Using Normal Flow and Texture Regularity. In *Pattern Recognition and Image Analysis*; Marques, J.S., Pérez de la Blanca, N., Pina, P., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2005; Volume 3523, pp. 223–230. [CrossRef]
10. Banerjee, T.; Keller, J.M.; Skubic, M.; Stone, E. Day or Night Activity Recognition From Video Using Fuzzy Clustering Techniques. *IEEE Trans. Fuzzy Syst.* **2014**, *22*, 483–493. [CrossRef]
11. Yan, Y.; Ricci, E.; Liu, G.; Sebe, N. Egocentric Daily Activity Recognition via Multitask Clustering. *IEEE Trans. Image Process.* **2015**, *24*, 2984–2995. [CrossRef]
12. Lin, W.; Chu, H.; Wu, J.; Sheng, B.; Chen, Z. A Heat-Map-Based Algorithm for Recognizing Group Activities in Videos. *IEEE Trans. Circuits Syst. Video Technol.* **2013**, *23*, 1980–1992. [CrossRef]
13. Chen, D.-Y.; Huang, P.-C. Motion-based unusual event detection in human crowds. *J. Vis. Commun. Image Represent.* **2011**, *22*, 178–186. [CrossRef]
14. Liu, C.; Yuen, P.C.; Qiu, G. Object motion detection using information theoretic spatio-temporal saliency. *Pattern Recognit.* **2009**, *42*, 2897–2906. [CrossRef]
15. Tanberk, S.; Kilimci, Z.H.; Tukel, D.B.; Uysal, M.; Akyokus, S. A Hybrid Deep Model Using Deep Learning and Dense Optical Flow Approaches for Human Activity Recognition. *IEEE Access* **2020**, *8*, 19799–19809. [CrossRef]
16. Cho, J.; Lee, M.; Chang, H.J.; Oh, S. Robust action recognition using local motion and group sparsity. *Pattern Recognit.* **2014**, *47*, 1813–1825. [CrossRef]
17. Yamato, J.; Ohya, J.; Ishii, K. Recognizing human action in time-sequential images using hidden Markov model. In Proceedings of the 1992 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Champaign, IL, USA, 15–18 June 1992. [CrossRef]
18. Bobick, A.; Davis, J. The recognition of human movement using temporal templates. *IEEE Trans. Pattern Anal. Mach. Intell.* **2001**, *23*, 257–267. [CrossRef]
19. 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]
20. Seo, H.J.; Milanfar, P. Action Recognition from One Example. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *33*, 867–882. [CrossRef]
21. Suk, H.-I.; Jain, A.K.; Lee, S.-W. A Network of Dynamic Probabilistic Models for Human Interaction Analysis. *IEEE Trans. Circuits Syst. Video Technol.* **2011**, *21*, 932–945. [CrossRef]
22. Turaga, P.; Chellappa, R.; Subrahmanian, V.S.; Udrea, O. Machine Recognition of Human Activities: A Survey. *IEEE Trans. Circuits Syst. Video Technol.* **2008**, *18*, 1473–1488. [CrossRef]
23. Hu, W.; Xie, N.; Li, L.; Zeng, X.; Maybank, S. A Survey on Visual Content-Based Video Indexing and Retrieval. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **2011**, *41*, 797–819. [CrossRef]
24. Fazli, M.; Kowsari, K.; Gharavi, E.; Barnes, L.; Doryab, A. HHAR-net: Hierarchical Human Activity Recognition using Neural Networks. In *Intelligent Human Computer Interaction—IHCI 2020*; Springer: Cham, Switzerland, 2020; pp. 48–58. [CrossRef]
25. Ferrari, A.; Micucci, D.; Mobilio, M.; Napoletano, P. On the Personalization of Classification Models for Human Activity Recognition. *IEEE Access* **2020**, *8*, 32066–32079. [CrossRef]
26. Jaouedi, N.; Boujnah, N.; Bouhlel, M.S. A new hybrid deep learning model for human action recognition. *J. King Saud Univ. Comput. Inf. Sci.* **2020**, *32*, 447–453. [CrossRef]
27. Andrade-Ambriz, Y.A.; Ledesma, S.; Ibarra-Manzano, M.-A.; Oros-Flores, M.I.; Almanza-Ojeda, D.-L. Human activity recognition using temporal convolutional neural network architecture. *Expert Syst. Appl.* **2022**, *191*, 116287. [CrossRef]
28. Khan, I.U.; Afzal, S.; Lee, J.W. Human Activity Recognition via Hybrid Deep Learning Based Model. *Sensors* **2022**, *22*, 323. [CrossRef]
29. Xu, Y.; Qiu, T.T. Human Activity Recognition and Embedded Application Based on Convolutional Neural Network. *J. Artif. Intell. Technol.* **2021**, *1*, 51–60. [CrossRef]

30. 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]
31. Hinton, G.E.; Osindero, S.; Teh, Y.-W. A Fast Learning Algorithm for Deep Belief Nets. *Neural Comput.* **2006**, *18*, 1527–1554. [CrossRef]
32. Lecun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
33. Lin, C.-M.; Tsai, C.-Y.; Lai, Y.-C.; Li, S.-A.; Wong, C.-C. Visual Object Recognition and Pose Estimation Based on a Deep Semantic Segmentation Network. *IEEE Sens. J.* **2018**, *18*, 9370–9381. [CrossRef]
34. Young, T.; Hazarika, D.; Poria, S.; Cambria, E. Recent Trends in Deep Learning Based Natural Language Processing. *IEEE Comput. Intell. Mag.* **2018**, *13*, 55–75. [CrossRef]
35. Xue, H.; Liu, Y.; Cai, D.; He, X. Tracking people in RGBD videos using deep learning and motion clues. *Neurocomputing* **2016**, *204*, 70–76. [CrossRef]
36. Minaee, S.; Boykov, Y.Y.; Porikli, F.; Plaza, A.J.; Kehtarnavaz, N.; Terzopoulos, D. Image Segmentation Using Deep Learning: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 3523–3542. [CrossRef] [PubMed]
37. Wang, W.; Zhou, T.; Porikli, F.; Crandall, D.; Van Gool, L. A survey on deep learning technique for video segmentation. *arXiv* **2021**, arXiv:210701153.
38. Tian, C.; Fei, L.; Zheng, W.; Xu, Y.; Zuo, W.; Lin, C.-W. Deep learning on image denoising: An overview. *Neural Netw.* **2020**, *131*, 251–275. [CrossRef]
39. Yu, S.; Ma, J.; Wang, W. Deep learning for denoising. *Geophysics* **2019**, *84*, V333–V350. [CrossRef]
40. Soleimani, E.; Nazerfard, E. Cross-subject transfer learning in human activity recognition systems using generative adversarial networks. *Neurocomputing* **2021**, *426*, 26–34. [CrossRef]
41. Dollár, P.; Rabaud, V.; Cottrell, G.; Belongie, S. Behavior recognition via sparse spatio-temporal features. In Proceedings of the 2005 IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, Beijing, China, 15–16 October 2005; IEEE: Piscataway, NJ, USA, 2005; pp. 65–72.
42. Rapantzikos, K.; Avrithis, Y.; Kollias, S. Dense saliency-based spatiotemporal feature points for action recognition. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; IEEE: Piscataway, NJ, USA, 2009; pp. 1454–1461.
43. Cao, L.; Liu, Z.; Huang, T.S. Cross-dataset action detection. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; IEEE: Piscataway, NJ, USA, 2010; pp. 1998–2005.
44. Li, C.; Zhang, B.; Chen, C.; Ye, Q.; Han, J.; Guo, G.; Ji, R. Deep Manifold Structure Transfer for Action Recognition. *IEEE Trans. Image Process.* **2019**, *28*, 4646–4658. [CrossRef]
45. Zhang, J.; Shum, H.P.H.; Han, J.; Shao, L. Action Recognition From Arbitrary Views Using Transferable Dictionary Learning. *IEEE Trans. Image Process.* **2018**, *27*, 4709–4723. [CrossRef]
46. Zhang, B.; Yang, Y.; Chen, C.; Yang, L.; Han, J.; Shao, L. Action Recognition Using 3D Histograms of Texture and a Multi-Class Boosting Classifier. *IEEE Trans. Image Process.* **2017**, *26*, 4648–4660. [CrossRef]
47. Poppe, R. A survey on vision-based human action recognition. *Image Vis. Comput.* **2010**, *28*, 976–990. [CrossRef]
48. Rogez, G.; Weinzaepfel, P.; Schmid, C. LCR-Net: Localization-ClassificationRegression for Human Pose. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 3433–3441.
49. Zhou, X.; Huang, Q.; Sun, X.; Xue, X.; Wei, Y. Towards 3D Human Pose Estimation in the wild: A weakly-supervised approach. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 398–407.
50. Katircioglu, I.; Tekin, B.; Salzmann, M.; Lepetit, V.; Fua, P. Learning Latent Representations of 3D Human Pose with Deep Neural Networks. *Int. J. Comput. Vis.* **2018**, *126*, 1326–1341. [CrossRef]
51. Kibet, D.; So, M.S.; Kang, H.; Han, Y.; Shin, J.-H. Sudden Fall Detection of Human Body Using Transformer Model. *Sensors* **2024**, *24*, 8051. [CrossRef] [PubMed]
52. Guo, Z.; Lu, M.; Han, J. Temporal Graph Attention Network for Spatio-Temporal Feature Extraction in Research Topic Trend Prediction. *Mathematics* **2025**, *13*, 686. [CrossRef]
53. Tsai, J.-K.; Hsu, C.-C.; Wang, W.-Y.; Huang, S.-K. Deep Learning-Based Real-Time Multiple-Person Action Recognition System. *Sensors* **2020**, *20*, 4758. [CrossRef]
54. Gopalakrishnan, T.; Wason, N.; Krishna, R.J.; Krishnaraj, N. Comparative Analysis of Fine-Tuning I3D and SlowFast Networks for Action Recognition in Surveillance Videos. *Eng. Proc.* **2023**, *59*, 203.
55. Vishwakarma, D.; Kapoor, R. Hybrid classifier based human activity recognition using the silhouette and cells. *Expert Syst. Appl.* **2015**, *42*, 6957–6965. [CrossRef]

56. Yang, J.; Nguyen, M.N.; San, P.P.; Li, X.; Krishnaswamy, S. Deep convolutional neural networks on multichannel time series for human activity recognition. In Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, Buenos Aires, Argentina, 25–31 July 2015; pp. 3995–4001.
57. Wang, A.; Chen, G.; Yang, J.; Zhao, S.; Chang, C.-Y. A Comparative Study on Human Activity Recognition Using Inertial Sensors in a Smartphone. *IEEE Sens. J.* **2016**, *16*, 4566–4578. [CrossRef]
58. Vishwakarma, D.K.; Singh, K. Human Activity Recognition Based on Spatial Distribution of Gradients at Sublevels of Average Energy Silhouette Images. *IEEE Trans. Cogn. Dev. Syst.* **2017**, *9*, 316–327. [CrossRef]
59. Inoue, M.; Inoue, S.; Nishida, T. Deep recurrent neural network for mobile human activity recognition with high throughput. *Artif. Life Robot.* **2018**, *23*, 173–185. [CrossRef]
60. Phyo, C.N.; Zin, T.T.; Tin, P. Deep Learning for Recognizing Human Activities Using Motions of Skeletal Joints. *IEEE Trans. Consum. Electron.* **2019**, *65*, 243–252. [CrossRef]
61. Zhu, R.; Xiao, Z.; Li, Y.; Yang, M.; Tan, Y.; Zhou, L.; Lin, S.; Wen, H. Efficient Human Activity Recognition Solving the Confusing Activities Via Deep Ensemble Learning. *IEEE Access* **2019**, *7*, 75490–75499. [CrossRef]
62. Dua, N.; Singh, S.N.; Semwal, V.B. Multi-input CNN-GRU based human activity recognition using wearable sensors. *Computing* **2021**, *103*, 1461–1478. [CrossRef]
63. Rahayu, E.S.; Yuniarno, E.M.; Purnama, I.K.E.; Purnomo, M.H. A Combination Model of Shifting Joint Angle Changes with 3D-Deep Convolutional Neural Network to Recognize Human Activity. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2024**, *32*, 1078–1089. [CrossRef]
64. Song, Z.; Zhao, P.; Wu, X.; Yang, R.; Gao, X. An Active Control Method for a Lower Limb Rehabilitation Robot with Human Motion Intention Recognition. *Sensors* **2025**, *25*, 713. [CrossRef]
65. Bsoul, A.A.R.K. Human Activity Recognition Using Graph Structures and Deep Neural Networks. *Computers* **2024**, *14*, 9. [CrossRef]
66. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef]
67. Vrigkas, M.; Nikou, C.; Kakadiaris, I.A. A Review of Human Activity Recognition Methods. *Front. Robot. AI* **2015**, *2*, 28. [CrossRef]
68. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
69. Yu, K.; Lin, Y.; Lafferty, J. Learning image representations from the pixel level via hierarchical sparse coding. In Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Colorado Springs, CO, USA, 20–25 June 2011; IEEE: Piscataway, NJ, USA, 2011; pp. 1713–1720.
70. Leung, T.; Malik, J. Representing and Recognizing the Visual Appearance of Materials using Three-dimensional Textons. *Int. J. Comput. Vis.* **2001**, *43*, 29–44. [CrossRef]
71. Rubner, Y.; Tomasi, C.; Guibas, L.J. The Earth Mover's Distance as a Metric for Image Retrieval. *Int. J. Comput. Vis.* **2000**, *40*, 99–121. [CrossRef]
72. Ling, H.; Okada, K. Diffusion Distance for Histogram Comparison. In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition—Volume 1 (CVPR'06), New York, NY, USA, 17–22 June 2006; IEEE: Piscataway, NJ, USA, 2006; pp. 246–253.
73. Paramasivam, K.; Sindha, M.M.R.; Balakrishnan, S.B. KNN-Based Machine Learning Classifier Used on Deep Learned Spatial Motion Features for Human Action Recognition. *Entropy* **2023**, *25*, 844. [CrossRef] [PubMed]
74. Ikizler, N.; Duygulu, P. Histogram of oriented rectangles: A new pose descriptor for human action recognition. *Image Vis. Comput.* **2009**, *27*, 1515–1526. [CrossRef]
75. Gorelick, L.; Blank, M.; Shechtman, E.; Irani, M.; Basri, R. Actions as Space-Time Shapes. *IEEE Trans. Pattern Anal. Mach. Intell.* **2007**, *29*, 2247–2253. [CrossRef] [PubMed]
76. Schuldt, C.; Laptev, I.; Caputo, B. Recognizing human actions: A local SVM approach. In Proceedings of the International Conference on Pattern Recognition, Cambridge, UK, 23–26 August 2004; IEEE: Piscataway, NJ, USA, 2004; Volume 3, pp. 32–36.
77. Ryoo, M.S.; Chen, C.C.; Aggarwal, J.K.; Roy-Chowdhury, A. *An Overview of Contest on Semantic Description of Human Activities (SDHA) 2010*; ICPR 2010. Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2010; Volume 6388. [CrossRef]
78. Liu, J.; Luo, J.; Shah, M. Recognizing realistic actions from videos 'in the wild'. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; IEEE: Piscataway, NJ, USA, 2009; pp. 1996–2003.
79. Guo, K.; Ishwar, P.; Konrad, J. Action Recognition From Video Using Feature Covariance Matrices. *IEEE Trans. Image Process.* **2013**, *22*, 2479–2494. [CrossRef]
80. Wong, S.-F.; Cipolla, R. Extracting Spatiotemporal Interest Points using Global Information. In Proceedings of the 2007 IEEE 11th International Conference on Computer Vision, Rio de Janeiro, Brazil, 14–21 October 2007; IEEE: Piscataway, NJ, USA, 2007; pp. 1–8.

81. Ali, S.; Shah, M. Human Action Recognition in Videos Using Kinematic Features and Multiple Instance Learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2008**, *32*, 288–303. [CrossRef]
82. Ikizler-Cinbis, N.; Sclaroff, S. Object, scene and actions: Combining multiple features for human action recognition. In Proceedings of the Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, 5–11 September 2010; Proceedings, Part I 11. Springer: Berlin/Heidelberg, Germany, 2010; pp. 494–507.
83. Le, Q.; Zou, W.; Yeung, S.; Ng, A. 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, Colorado Springs, CO, USA, 20–25 June 2011; IEEE: Piscataway, NJ, USA, 2011. [CrossRef]
84. Wang, H.; Klaser, A.; Schmid, C.; Liu, C.-L. Action recognition by dense trajectories. In Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Colorado Springs, CO, USA, 20–25 June 2011; IEEE: Piscataway, NJ, USA, 2011; pp. 3169–3176.
85. Noguchi, A.; Yanai, K. A surf-based spatio-temporal feature for feature-fusion-based action recognition. In Proceedings of the Trends and Topics in Computer Vision: ECCV 2010 Workshops, Heraklion, Crete, Greece, 10–11 September 2010; Revised Selected Papers, Part I 11. Springer: Berlin/Heidelberg, Germany, 2012; pp. 153–167.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.