Hindawi
Computational Intelligence and Neuroscience
Volume 2020, Article ID 8852404, 30 pages
https://doi.org/10.1155/2020/8852404

Research Article

A Novel Parameter Initialization Technique Using RBM-NN for Human Action Recognition

Deepika Roselind Johnson[1] **and V.Rhymend Uthariaraj**[2]

[1]*DCSE, CEG–Anna University, Guindy, Chennai, India*
[2]*RCC, CEG–Anna University, Guindy, Chennai, India*

Correspondence should be addressed to Deepika Roselind Johnson; deepikaroselind888@gmail.com

Received 13 May 2020; Revised 8 August 2020; Accepted 20 August 2020; Published 10 September 2020

Academic Editor: Raşit Köker

Human action recognition is a trending topic in the field of computer vision and its allied fields. The goal of human action recognition is to identify any human action that takes place in an image or a video dataset. For instance, the actions include walking, running, jumping, throwing, and much more. Existing human action recognition techniques have their own set of limitations when it concerns model accuracy and flexibility. To overcome these limitations, deep learning technologies were implemented. In the deep learning approach, a model learns by itself to improve its recognition accuracy and avoids problems such as gradient eruption, overfitting, and underfitting. In this paper, we propose a novel parameter initialization technique using the Maxout activation function. Firstly, human action is detected and tracked from the video dataset to learn the spatial-temporal features. Secondly, the extracted feature descriptors are trained using the RBM-NN. Thirdly, the local features are encoded into global features using an integrated forward and backward propagation process via RBM-NN. Finally, an SVM classifier recognizes the human actions in the video dataset. The experimental analysis performed on various benchmark datasets showed an improved recognition rate when compared to other state-of-the-art learning models.

1. Introduction

Human action recognition [1] is used for a variety of applications such as video surveillance [2], retrieval [3, 4], and detection [5–7]. The action recognition is performed by computational algorithms [8–10] that understand and detect human actions. These computational algorithms generate a label after detecting a human action. Action recognition involves extracting and learning human actions [11–13]. It can be performed by using three techniques—traditional design features, deep learning, and hybrid extraction [14]. Among these techniques, the hybrid extraction technique [15] has gained prominence in recent years. It involves using both traditional and deep learning techniques for recognition.

In traditional methods [16–20], artificial actions such as spatial convolutions [21, 22], temporal convolutions, and fusion techniques are used for extraction and recognition. Though they provide a good recognition rate, there have been no recent advances. Action recognition is comprised of two components: representation [23–27] and classification [25]. The human actions in a video sequence are generated as a space-time feature in 3D representation [28, 29]. They are comprised of both spatial and dynamic information; the spatial information includes human pose, and dynamic information includes motion. The movement is captured through anchors or bounding boxes to detect the subject from cluttered backgrounds. To capture the spatial-temporal features in human actions, various methods use Poisson distribution to extract the shape features [30, 31]. For action representation and classification, the spatial-temporal information is taken as input. The spatial-temporal saliency is computed from the moving parts and the local orientation is determined. These local representations are converted into global features by computing the weighted average of each point inside the bounding box and analyzing the different geometrical properties [32, 33].

Initially, the spatial-temporal points were extracted using Laptev's [23] and Harris corner detector [24] in the spatial-temporal domain. Gaussian kernel [34] is applied to the video sequence to obtain a response function for the spatial-temporal dimensions. Other prominent methods such as 2D Gaussian smoothing [35] were applied for obtaining the spatial features, and 1D Gabor filter is applied for obtaining the temporal features along with other information such as raw pixels, gradient, and flow features. Principal component analysis [36–38] is applied to the vector features for dimensionality reduction. The detection algorithms such as 3D SFIT [39], HOG3D [7, 40], HOG [41], and HOF [41] are used for describing the trajectories [42–44].

The spatial-temporal point of interest [45] captures only short-term distance. However, to describe the change in motion, it is necessary to track the points continuously. The trajectories along with the interest points are detected and tracked using Haris3D [24] with the KLT tracker [46]. Using this method [47], the trajectories are mapped with corresponding SIFT points over consecutive frames. Using the HOG, HOF, and MBH [48] features, the intertrajectories and intratrajectories are described. After the action is represented, action classifiers [30, 31, 45, 49–51] are applied to the training samples to determine the class boundaries. The human actions are classified into two types: direct classification and sequential method. The direct classification involves the extraction of a feature vector and recognition of actions from classifiers using SVM [36] and K-NN method [52, 53]. In the sequential method, the temporal features such as appearance and pose are obtained from the hidden Markov model [54–56], conditional random fields [57–60], and structured support vector machine [61–64]. Furthermore, representative key poses are learned for efficient representation of human actions [33, 34, 65–72] to build a compact pose sequence.

Deep learning techniques [73] such as 2D ConvNets [21, 74] and 3D ConvNets [26] perform feature learning via convolution operator and temporal modeling [75]. The initialization of a deep neural network [72] is crucial for training the model. To ensure that the state of the hidden layers follow a uniform distribution, a model parameter [76–78] is initialized. If the model parameter [79, 80] is not properly initialized, it leads to gradient explosion. The most commonly used technique is the Xavier initialization method [81] modeled based on the sigmoid activation function. Many models use ReLU activation function [82], RBMs [83, 84], and other methods [85] for learning.

In this paper, we propose a novel parameter initialization technique using the Maxout activation function (MAF) via restricted Boltzmann machine-neural network (RBM-NN).

The spatial and temporal features required for human action recognition are obtained from the video sequence via a feature learning process. The extracted spatial and temporal features are trained using RBM-NN. The RBM-NN converts the local features into global features using an integrated forward and backward propagation process. An SVM classifier is used for recognizing the human actions in the video sequence.

Section 2 describes the process of tracking human action from video sequences, extraction of shape features, and construction of an RBM-NN. Section 3 describes parameter initialization using an activation function, forward propagation, backward propagation, and action recognition using an SVM classifier. Section 4 consists of data preprocessing and model training for analyzing the effectiveness of the parameter initialization technique. Section 5 discusses the experimentation setup, result analysis performed on various benchmark datasets, influence of the learning parameter on model accuracy, and the loss function. Finally, Section 6 consists of concluding remarks followed by references.

2. Methodology

The spatial-temporal features [86, 87] for human action recognition are performed via a feature learning process [59, 62], as shown in Figure 1. The first step involves using detection and sequence tracking algorithm [88] to identify human action features. Secondly, the action tracking sequence is segregated into blocks to extract the shape features using the neural network layers implemented by RBM [83, 89]. The model is implemented by dividing the network layers and feeding the output of the first layer as input to the second layer to learn the spatial-temporal features. The second hidden layer is used for dimensionality reduction of the output from the first layer and to reduce computational efficiency.

2.1. Human Action Tracking from Video Sequence. The action changes in the human body are detected from video frames by posture and action changes. Target detection and tracking such as pedestrian detection algorithm [90, 91] are used to automatically detect and track the action sequences. A bounding box tracks the subject of interest and is optimized based on pose normalization. From the video dataset, the length of the tracking sequence is set to a fixed length L. If the length of the initial tracking sequence is greater than L, the redundant frames are discarded. If the length of the initial tracking sequence is lesser than L, the tracking sequence is extended by the zero-padding method and is set to L frames. The human actions from the tracking sequence are denoted by a_i, and other actions are denoted as o_i.

2.2. Extracting Shape Features. Every tracking sequence is divided into video blocks, and the initialization parameters are specified as $vb_w \times vb_h$. The segregated blocks are denoted as $V_k, k \in K$, where $K = \left\{1, 2, \ldots, V_k^{vb_w \times vb_h}\right\}$ corresponds to the spatial position of the block. In the proposed method, a deep neural network is used for extracting the spatial-temporal features from low-level features. The first step involves segregating blocks into individual frames $B_k, k \in K, B_n^k$, where $n = 1, 2, \ldots, L$ into grid cells $C_w \times C_h$. Each grid cell is computed in C_d directions in the histogram of oriented gradients (HOGs) and represents the shape characteristics. The shape dimensions of each image frame are denoted as $S_w \times S_h \times S_w$. The feature vector is represented as $(s_{m1}^k, s_{m2}^k, \ldots, s_{nm}^k)$, where $m = S_w \times S_h \times S_w$. The

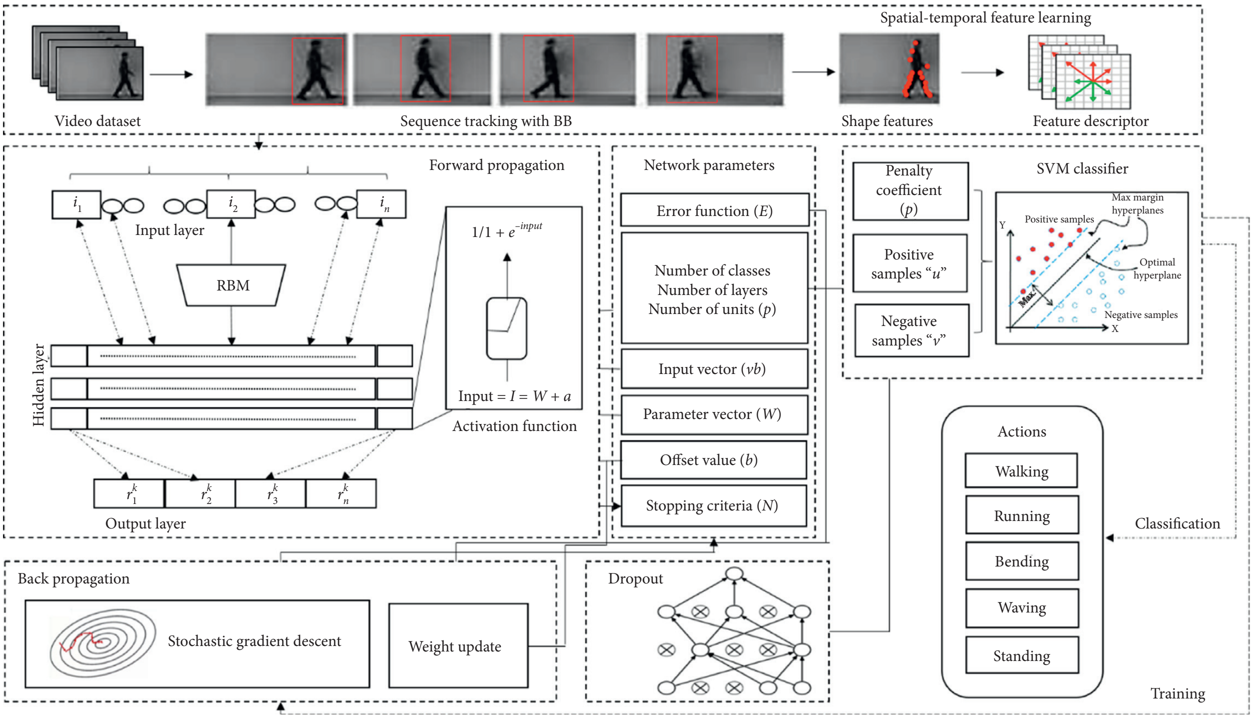


Figure 1: Proposed methodology for human action recognition.

initial component of the shape feature of the image frame IF^k_m is indicated as s^k_{nt}, where $t = 1, 2, \ldots, n$. The shape features from each block are extracted and divided into a long vector. These individual feature vectors represent the shape features.

During action recognition, the pose of the person is estimated and the shape features are extracted from the tracking sequence. The extracted shape features, i.e., pose in individual frames are normalized. The frame from a tracking sequence is represented as $V^1_k, V^2_k, \ldots, V^{vb_w \times vb_h}_k$, where $k = 1, 2, \ldots, n$. The normalized shape vectors for every frame in the tracking sequence are given as

$$I^k_{nt} = \frac{s^k_{nt}}{\left(\sum_{k=1}^{vb_w \times vb_h} \sum_{l=1}^{s} \left|s^k_{nl}\right|^2\right)^{(1/2)}}, \tag{1}$$

where $1 \leq t \leq s$, I^k_{nt} is the normalized shape feature vector and the component s^k_{nt} is the shape factor vector that corresponds to the normalized value. The shape feature for every individual frame in the block is denoted as $B^k_n = (l^k_{n1}, l^k_{n2}, \ldots, l^k_{nm})$, where $k \in K, 1 \leq n \leq L$. The shape features from the video block are represented as $B^k_1, B^k_2, \ldots, B^k_L$. The dimensional features are represented as $L \times S_w \times S_h \times S_w \cdot l^k_{nt} \in [0, 1]$. The eigenvectors of the shape features are denoted as $B^k_1, B^k_2, \ldots, B^k_L$ and is provided as input to train the RBM-NN.

2.3. Constructing an RBM-Neural Network. Restricted Boltzmann machine [54, 63] is comprised of a network architecture that consists of two neuron layers: the input layer and the hidden layer. The nodes present in the input layer and hidden layers are connected, but they are connected with a particular layer. RBMs are capable of self-learning through discrete distribution via the hidden neutrons. The input layer consists of multiple RBMs, as shown in Figure 2, to describe the distribution of action characteristics. For each type of action category, the training samples are fed to the RBMs with spatial features.

The output layers from each RBM comprise of N neurons, and the value of N has a direct influence on the distribution of every action learned. The proposed method analyses influence that the value N has on the experimental results. For every RBM present in the neuron network layer, the limits are set as $k = 1, \ldots, vb_w \times vb_h$. It is used for training the various shape features from the blocks along with their corresponding spatial position $'k'$ as input. The input video block has the following shape feature $I^k = (i^k_{11}, i^k_{12}, \ldots, i^k_{Ln})^L$, and the corresponding output is represented as $R_k = R_k = (r^k_1, r^k_2, \ldots, r^k_N)^L$. The restrictions in the RBM-NN, its state, and energy of the neurons $\{I_k,\}$ is defined as

$$\begin{aligned} E\left(I^k, R^k; \theta^k\right) &= -\left(I^k\right)^L P^k R^k - \left(b^k\right)^L I^k - \left(a^k\right)^L R^k \\ &= -\sum_{n=1}^{L\times m} \sum_{t=1}^{T} P^k_{nt} I^k_n R^k_t - \sum_{n=1}^{L\times m} b^k_n I^k_n - \sum_{t=1}^{T} a^k_t R^k_t, \end{aligned} \tag{2}$$

where $\theta^k = \{P^k\}$ in which θ^k is the RBM parameter and P^k represents the symmetric correlation between the input and output neurons. Also, a_k and b_k indicate the deviation among the column vectors generated in the input and the output layer. The set of model parameters used in RBM is learned using the contrastive divergence (CD) algorithm

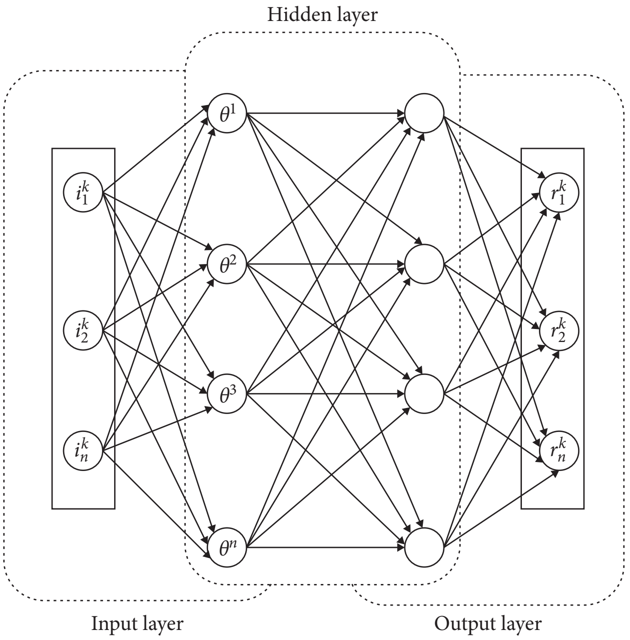


Figure 2: Multiple restricted Boltzmann machines.

[92]. The CD algorithm is effective for training undirected graphical models (RBMs) and estimates the energy gradient given a set of model parameters along with the training data. The CD provides the gradient estimates and enables the model to keep balanced and avoids the issue of gradient explosion and overfitting. The distribution between the input and output neurons for a single RBM is given as

$$G\left(I^k, R^k; \theta^k\right) = \frac{1}{Z\left(\theta^k\right)} \exp\left(-E\left(I^k, R^k; \theta^k\right)\right). \tag{3}$$

$$Z\left(\theta^k\right) = \sum_{I^k} \sum_{R^k} \exp\left(-E\left(I^k, R^k; \theta^k\right)\right), \tag{4}$$

where θ^k is the partition function and the conditional probability distribution is derived from equation (3):

$$\begin{aligned} g\left(R_t^k = 1 \,\middle|\, I^k\right) &= h\left(\sum_n W_{nt}^k I_n^k + a_t^k\right), \\ g\left(I_N^k = 1 \,\middle|\, R^k\right) &= h\left(\sum_t W_{nt}^k R_n^k + b_t^k\right). \end{aligned} \tag{5}$$

The proposed method trains the RBM in the first layer of the neural network architecture. The network parameter set of the multiple RBM neural network layers for every action is denoted as $\theta = \theta^1, \theta^2, \ldots, \theta^n$.

The proposed work is used for training the two-layer neural network for every action category. The second layer of the neural network is also an individual RBM and solely used for dimensionality reduction of the output obtained from the first layer. The parameter of the network layer is denoted as (W, a). For every action category, the input from an action sequence will provide the feature vectors as output.

The output of the trained two-layered neural network is modeled based on spatial-temporal shape feature learning from the block. The spatial-temporal individualities are represented as $R = (r_1, r_2, \ldots, r_A)$, where A is the set generated based on experience and is denoted as $A = 16 \times vb_w \times vb_h \times N$.

3. Parameter Initialization Using Activation Function

3.1. Importance of Effective Parameter Initialization. To build an efficient model for human action recognition, an RBM-NN architecture is defined in the proposed work and it is trained to learn the parameters. The RBM-NN architecture is trained using the following steps: parameter initialization, optimization algorithm, forward propagation, cost function computation, gradient cost computation using back propagation, and parameter updation.

When testing data are provided, the network uses the trained model to predict the class. For a network to perform efficiently, it is crucial to initialize the right parameter to avoid the problem of gradient explosion and vanishing.

Case 1. If the initialized parameter is large, it leads to a gradient explosion:

$$\text{initialized weight} \gg \text{identity matrix}. \tag{6}$$

Case 2. If the initialized parameter is small, it leads to vanishing gradients:

$$\text{initialized weight} \ll \text{identity matrix}. \tag{7}$$

To prevent the problem specified above, a set of rules have to be adhered to while initializing the network parameter. Initially, the mean value of the activation function must always be zero. Finally, the variance of the activation function must remain uniform throughout the network layers. If the rules are not followed, it gives rise to a locally optimal solution which renders the model untrainable and improper feature extraction.

The model parameter is initialized based on two categories: parameter initialization by pretraining a model and parameter optimization by training the neural network. In the first method, a model is trained using the unsupervised model, and an AutoEncoder [93] is used to build a layer-by-layer unsupervised objective function. The layer-by-layer training is performed on equal depth neural networks to obtain the feature representations from the input. Pretraining a model involves computational overhead, and the training efficiency is affected. The second method involves initializing the parameter and its optimization using neural networks. The parameter can be initialized using a nonlinear activation function and backpropagation.

3.2. Parameter Initialization Using Maxout Activation Function. In this paper, the parameter initialization technique is modeled using a Maxout layer. The layer consists of an activation function which takes the maximum of the inputs. When compared to other activation functions, Maxout activation function [94] performs well due to the dropout technique. Dropout is a model averaging technique where a random subnetwork is trained for every iteration and the weights are averaged at the end. An approximation has to be used as these weights cannot be averaged explicitly. The inputs to the Maxout layer are not dropped using the corresponding activation function. The input with the maximum value for the data point is not affected as the dropout occurs in the linear part. Thus, it leads to efficient model averaging as the averaging approximation is for linear networks.

In the proposed work, it is assumed that the state of the neuron node follows a uniform distribution required for a Maxout activation function. It is an activation function that is capable of training itself in our model. It performs a piecewise linear approximation on ReLU, absolute function, and quadratic function to a random convex function. It considers the maximum value from a set of linear values that are determined beforehand. The Maxout implements ReLU and absolute function using two linear functions and the quadratic function using four linear functions. It can approximate any function using multiple linear functions and is known as piece-wise linear approximation.

The Maxout unit is implemented using the following function:

$$f(x) = \max(w_1 x + b_1, w_2 x + b_2, \ldots, w_n x + b_n), \tag{8}$$

where n is the number of linear combinations. If w_1 is set to one, all the other values take the value zero such that the proposed activation function becomes equivalent to the traditional activation functions.

As mentioned earlier, any continuous piece-wise linear approximation can be expressed as a difference between two convex functions:

$$g(x) = f_1(x) - f_2(x), \tag{9}$$

where $f_1(x)$ and $f_2(x)$ are the convex functions and $g(x)$ is a continuous piece-wise linear approximation function. From equation (9), it can be deduced that a Maxout layer comprising two Maxout units can be used to approximate any continuous function randomly.

Also, both ReLU and leaky ReLU are considered to be special cases of a Maxout unit and enjoy all the benefits of a ReLU unit. It implements linearity of operations with no saturation and avoids the issue of dying ReLU. A Maxout can be formed with more units, but this will increase the capacity of the network and requires more training. Thus, Maxout units are considered as universal approximators.

The MAF is modeled based on theoretical derivation for parameter initialization of the model. Both forward propagation and backward propagation process in the network are analyzed to ensure that every neuron follows a uniform distribution.

3.3. Forward Propagation Process. To perform forward propagation, the following assumptions are made: (1) the input vector *vb* and the parameter vector *W* must be independent; (2) the input vector *vb* and the parameter vector *W* must follow the same distribution; (3) the initial distribution of the parameter vector *W* must be symmetrical about the zero-point; and (4) the offset value *b* of each layer must always be zero.

The response of the hidden convolution layer in the RBM-NN is given as

$$z_t = x_t^L W_t + b_t, \tag{10}$$

where t denotes the n^{th} hidden layer of the RBM-NN, among which $x_t \in A_p$, x^t is the original input vector, and the mean value is set to zero after processing.

$$p = u^2 i, \tag{11}$$

where p is the number of input nodes connected to one neuron node, u is the size of the convolution kernel, and $'i'$ is the number of input channels to the model. The output of every neuron node is passed through the MAF provided as follows:

$$f(x) = \max(w_1 x + b_1, w_2 x + b_2, \ldots, w_n x + b_n), \tag{12}$$

where n is the number of linear combinations. If w_1 is set to one, all the other values take the value zero such that the proposed activation function becomes equivalent to the traditional activation functions. The problem of local linearity in the proposed activation function eliminates the issue of gradient explosion, but there is an increase in computational overhead during the training process.

The variance of the initialization parameter can be obtained as follows:

$$\text{Var}[z_t] = p_t \text{Var}[W_t x_t]. \tag{13}$$

The weight W_t and hidden layers have to adhere to Gaussian distribution with a mean value of zero as per assumptions 2 and 3. The initial state and the parameter vectors are assumed to be independent of each other as per assumption 1. Thus, the variance in the initialization parameter is provided:

$$\text{Var}[z_t] = p_t \text{Var}[W_t] E[x_t^2], \tag{14}$$

where $E[x_t^2]$ is the exception function. The proposed activation function can be simplified by considering two linear functions given as follows:

$$x_t = r_{t-1}(x_{t-1}) = \max(z_{t-1,1}, z_{t-1,2}). \tag{15}$$

Based on assumption 4, the offset value b_{t-1} is always set to zero and the mean weights W_t are also set to zero. The values $z_{t-1,1}, z_{t-1,2}$ are assumed to be symmetrical at the mean point and follow the same distribution.

The expectation function $E[x_t^2]$ and the variance $\text{Var}[z_{t-1}]$ are defined as follows:

$$x_t = \frac{z_{t-1,1} + z_{t-1,2} + |z_{t-1,1} - z_{t-1,2}|}{2}. \tag{16}$$

The expectation $E[x_t^2]$ value is obtained by substituting equation (15):

$$E\left[x_t^2\right] = \frac{1}{2}\left(\mathrm{Var}\left[z_{t-1,1}\right] + \mathrm{Var}\left[z_{t-1,2}\right]\right). \tag{17}$$

As per assumption 2, the values $z_{t-1,1}$ and $z_{t-1,2}$ follow the uniform distribution and the new variance is obtained as follows:

$$\mathrm{Var}\left[z_{t-1}\right] = \mathrm{Var}\left[z_{t-1,1}\right] = \mathrm{Var}\left[z_{t-1,2}\right]. \tag{18}$$

Substituting the variance value obtained from equation (17) into equation (16), we get

$$E\left[x_t^2\right] = \mathrm{Var}\left[z_{t-1}\right]. \tag{19}$$

The relationship between the variances is obtained by substituting equation (17) into equation (13) as follows:

$$\mathrm{Var}\left[z_t\right] = p_t \mathrm{Var}\left[W_t\right]\mathrm{Var}\left[z_{t-1}\right]. \tag{20}$$

The difference in variance between the first hidden layer and the last hidden layer is obtained as follows:

$$\mathrm{Var}\left[z_T\right] = \left(\prod_{t=2}^{T} p_t \mathrm{Var}\left[W_t\right]\right)\mathrm{Var}\left[z_t\right]. \tag{21}$$

The initialization parameter for a neural network model must follow the necessary condition:

$$p_t \mathrm{Var}\left[W_t\right] = 1, \quad \forall t. \tag{22}$$

When t is set to 1, equation (21) is satisfied without the interference on the input vector by the activation function. Based on the theoretical assumption, each node in the hidden layer behaves similarly to a neural network. Also, the model parameter initialization for every node in the hidden layer satisfies the Gaussian distribution.

3.4. Backpropagation Process

3.4. Backpropagation Process. In backpropagation, the following assumptions are made similar to forward propagation: (1) the gradient Δr_t and the parameter vector W must be independent of each other; (2) the gradient Δr_t and the parameter vector W must follow the same distribution; and (3) the gradient Δr_t and the parameter vector W must have zero symmetry for $E\left[\Delta x_t\right] = 0$.

The concentration of gradients obtained by the convolution parameter is shown as follows:

$$\Delta x_t = W_t \Delta^{\Delta} r_t, \tag{23}$$

where Δx_t and Δr_t are the gradients that represent the loss functions. The value of the activation function is obtained when $a = 0$:

$$\Delta z_t, n = f'\left(z_t, n\right)\Delta x_{t+1}, \quad n \in \{1, 2\}. \tag{24}$$

If $f'\left(z_t, n\right) = 1$ and $f'\left(z_t, n\right) = 0$, each has half probability of occurrence. Moreover, $f'\left(z_t, n\right) = 1$ and Δx_{t+1} are independent of each other based on assumption 1.

The initial condition $n \in \{1, 2\}$ is provided:

$$\begin{aligned} E\left[\Delta r_t\right] &= E\left[\Delta x_t, n\right], \\ E\left[\left(\Delta r_t\right)^2\right] &= \mathrm{Var}\left[\Delta r_t\right] = \frac{1}{2}\mathrm{Var}\left[\Delta x_{t+1}\right]. \end{aligned} \tag{25}$$

The variance function for the gradient is obtained as follows:

$$\mathrm{Var}\left[\Delta x_t\right] = \frac{1}{2} r\wedge_t \mathrm{Var}\left[W_t\right]\mathrm{Var}\left[\Delta x_{t+1}\right]. \tag{26}$$

The relationship between $\mathrm{Var}\left[\Delta x_2\right]$ and $\mathrm{Var}\left[\Delta x_{T+1}\right]$ can be defined as follows:

$$\mathrm{Var}\left[\Delta x_2\right] = \mathrm{Var}\left[\Delta x_{T+1}\right]\left(\prod_{t=2}^{T} \frac{1}{2} r\wedge_t \mathrm{Var}\left[W_t\right]\right). \tag{27}$$

For the gradient to move smoothly, the following initial condition has to be satisfied:

$$\frac{1}{2} r\wedge_t \mathrm{Var}\left[W_t\right] = 1, \quad \forall t \in [2, T]. \tag{28}$$

The parameter for neural network model W also follows the same distribution based on assumption 2:

$$W_t \sim N\left(0, \frac{2}{r\wedge_t}\right). \tag{29}$$

It is not possible to perform both forward and backward propagation at the same time. Thus, the parameter has to be optimized as follows:

$$\min_{\tau_t}\left(\tau_t - r_t\right)^2 + \left(\tau_t - \frac{1}{2} r\wedge_t\right)^2. \tag{30}$$

The optimized solution for the proposed initialization parameter for RBM-NN based on uniform distribution is obtained:

$$W_t \sim N\left(0, \frac{4}{2r_t + r\wedge_t}\right). \tag{31}$$

3.5. SVM Classifier for Action Recognition. An SVM classifier is built for each action category. The training of the RBM-NN is categorized into two samples: positive samples and negative samples. The samples which correspond to action categories a_i are classified as positive samples $'u'$ and other actions o_i as negative samples $'v'$. The parameter vector W and the other variables are optimized. If there is an imbalance in the positive and negative samples, the classification accuracy in the training phase is affected. To overcome the issue of accuracy, a penalty coefficient parameter $'P'$ is introduced. If the training set has less positive samples, a higher penalty coefficient P is enforced and the negative samples are introduced to a lesser penalty coefficient P.

The SVM objective function for our proposed method is defined as follows:

$$\begin{aligned} &\min_{\omega,\varepsilon} && \frac{1}{2}\|\omega\|^2 + P + \sum_{i=1}^{u}\varepsilon_i + P - \sum_{j=u+1}^{u+v}\varepsilon_j \\ &\text{s.t.} && y_i\left[\left(\omega^L R_i\right) + b\right] \geq 1 - \varepsilon_i, \quad i = 1, 2, \ldots, u+v, \\ & && \varepsilon \geq 0, \end{aligned} \tag{32}$$

where $i = 1, 2, \ldots, u + v$, R_i is the spatial-temporal feature of the i^{th} action sample and (R_i, y_i) is the input of the SVM classifier. Also, $u + v$ is the total number of training samples used for training the SVM classifier. The SVM classifier is trained for each action category and represented as an action model (θ, W, a, b) comprising two-layer RBM-NN for human action recognition.

4. Result Analysis and Discussion

The parameter initialization proposed in the paper is verified and analyzed on the MS-COCO [95], ImageNet [96], and CIFAR-100 [97] datasets respectively. The RBM-NN comprises four convolution layers for analysis along with the loss function. The loss function considered in the model is the logistic loss layer obtained after downsampling. To prevent overfitting, the dataset is separated into batches and trained as submodels. The parameter is initialized randomly, and the submodels are trained using the dropout technique by randomly setting the output nodes to zero before updating the training set. The dropout probability for the model validation is set as 50% to determine the classification error rates.

4.1. Data Preprocessing. The training data are preprocessed by applying global contrast normalization and zero component analysis whitening [98]. The GCN technique prevents the images from exhibiting various levels of contrast. The mean value is subtracted, and the image is rescaled such that the standard deviation across the pixels is constant. ZCA whitening process ensures that the average covariance between the whitened pixel and the original image is maximal. For instance, it makes the data less redundant by removing the neighboring correlations in adjacent pixels.

4.2. Model Training. The models were initially trained using the Xavier initialization method [81] for parameter initialization and the model parameters. The Xavier initialization method is chosen since it keeps the variance uniform across each network layer as per the assumptions followed during the forward propagation process. The initial and model parameters must follow a uniform distribution specified below:

$$W \sim U\left[-\frac{\sqrt{6}}{\sqrt{n_k + n_{k+1}}},\right]\frac{\sqrt{6}}{\sqrt{n_k + n_{k+1}}}, \tag{33}$$

where n_k is the number of input nodes and n_{k+1} are the number of output nodes. The datasets MS-COCO [95], ImageNet [96], and CIFAR-100 [97] were considered as input for the proposed parameter initialization method and also compared with parameter initialized via the Xavier model. The proposed parameter initialization method showed similar results in the classification accuracy of the activation function. The improvement in classification accuracy has been attributed to the fact that nodes and states of the various hidden layers follow the same distribution pattern and avoids the problem of gradient explosion.

The dataset ImageNet comprises a 1000-class image problem and required 120 epochs. The MS-COCO comprises 80 classes and required 64 epochs for training. The CIFAR-100 dataset is comprised of 100 classes and required 200 epochs for training. The model required more layers for analysis along with the introduction of convolution kernels. The deep neural network model was able to perform iteration for 500,000 times with a learning rate set to 0.1. However, it was found that the learning rate decreased with an increase in the number of iterations. The comparison of the test error rates between the proposed initialization method and the Xavier initialization method is provided in Table 1. The analysis shows that the error rates obtained from the proposed method showed better results for both small (MS-COCO) and large datasets (ImageNet and CIFAR-100).

The model parameters along with the slack variables are initialized and optimized by the objective function used by the SVM classifier. During the training process, it was noticed that there was an imbalance between the positive and negative samples.

For instance, there were fewer positive samples in the training set when compared to the negative samples. Thus, a higher penalty coefficient $'P'$ was introduced to the positive samples to balance the training samples.

5. Experimentation Setup and Analysis

The human action recognition using the proposed method is performed using the datasets specified in Table 2 along with their classes, modalities, and environment type. These benchmark datasets are comprised of actions performed in both simple and cluttered background scenes. The datasets are divided into training and testing sets. This discriminative action is used for segmentation to reduce the background correlation between the training and the testing set. The model is trained using small samples, and the data expansion method [108] is used increasing the number of video samples present in the training set.

Initially, the actions are detected from the video blocks to extract the spatial-temporal features. The features are fed to the RBMs for training along with suitable model parameters via forward and backward propagation process. The output from the RBMs is fed to the SVM classifier for human action recognition. During the experiment analysis performed on the dataset, the influence of the N parameter is analyzed along with the penalty coefficient P. The effect of the number of output neurons for each RBM is obtained by adjusting the value of the parameter N. The number N of the output neurons is influenced by the average recognition rate of the

Table 1: Comparison of test error rates of the initialization method.

Dataset	Test error rates	
	Xavier parameter initialization method (%)	Proposed parameter initialization method (%)
MS-COCO	10.25 ± 0.02	8.15 ± 0.09
CIFAR-100	19.38 ± 0.19	17.51 ± 0.22
ImageNet	23.19 ± 0.12	21.25 ± 0.17

Table 2: Action video datasets used in our proposed work.

Dataset	Year	Videos	Classes	Modality	Environment type
Weizmann [99]	2005	90	10	RGB	Controlled
CAVIAR [100]	2005	390	13	RGB	Controlled
UCF sports action [101]	2009	1,100	11	RGB	Uncontrolled
KTH [102]	2004	599	6	RGB	Controlled
CASIA [103]	2007	1446	8	RGB	Controlled
i3DPost [104]	2009	104	13	RGB	Controlled
JHMDB [105]	2011	316	12	RGB	Uncontrolled
UCF101 [106]	2012	13,320	101	RGB	Uncontrolled
HMDB51 [107]	2011	7000	51	RGB	Uncontrolled

action sequence. The value of N determines the number of spatial-temporal features based on RBM-NN.

The SVM classifier is used for action recognition of multiple types of actions. The SVM classifier model calculates the shape features of the video blocks for each action category. After the classification values are compared, the largest classification value is set as an action label for the test video sequence. The actions from the tracking sequence are detected from the action video.

The proposed algorithm operates on the image sequences with varied focus points, deep learning is used for learning all the features, and SVM classification is performed. The proposed action recognition feature is more specific than other methods. Finally, the model is compared with other state-of-the-art techniques to compare the classification accuracy rate of the model.

5.1. Weizmann Dataset. The Weizmann dataset [99] is made available by the Weizmann Institute of Science and consists of two datasets. The event-based analysis dataset consists of long sequences of around 6000 frames comprising various people. The actions are divided into four categories: running in place, walking, running, and waving. The ground truth dataset is action annotated for every frame and can be temporally segmented. The second dataset Weizmann actions as space-time shapes dataset was created for human action recognition systems that are suitable for spatial and temporal volumes. The videos were recorded on a simple background with nine persons performing ten actions. The human actions have been divided into ten categories such as walking, running, jumping, galloping, bending, one-hand waving, two-hands waving, jumping in place, jumping jacks, and skipping, as specified in Figure 3. It is a database of 91 low-resolution video sequences. The dataset comprising 91 video sequences is divided into 60 video samples for the training set and 31 action samples for the testing set.

During experimentation, every action in the tracking sequence was divided into 180×144 (25 fps) video blocks. The parameter N is set to 300, where N represents the number of output neurons of each RBM present in the first neural network layer. The proposed method is compared with the reference method [109]. For determining the SVM classifier, set the penalty coefficient $P = 10$, and other slack variables are determined by the objective function. The neural network parameters are obtained by adaptive matching with the processed image data. The proposed work correctly identifies the rotation action of the Weizmann actions as space-time shapes dataset such as walking, running, jumping, bending, waving, and skipping.

The proposed method is compared with the reference model [110] proposed by Haiam et al. They proposed a trajectory-based approach for human action recognition to obtain the temporal discriminative features. The trajectories are extracted by detecting the STIPs and matching them with the SIFT descriptors in the video frames. The trajectory points are represented using the bag of words (BoW) model. Finally, an SVM-based approach is used for action recognition. From the confusion matrix shown in Figure 4, it can be noticed that there are some confusions in some frames for actions such as walking, running, jumping, and skipping. Also, the action two-hand waving is similar to jumping jacks. These confusions influence the classification accuracy of the proposed model.

The proposed approach is evaluated with the classification accuracy obtained by the following descriptors: TD, HOG, HOF, MBH, and the combinations, as shown in Figure 5. Table 3 shows the average recognition rate for the dataset along with the reference method. It can be noticed that the accuracy rate for the HOG, HOF, and combined features achieved better accuracy when compared to the proposed method due to variations in the codebook sizes and model representation. The vector patches are converted to codewords to produce a codebook comprising similar

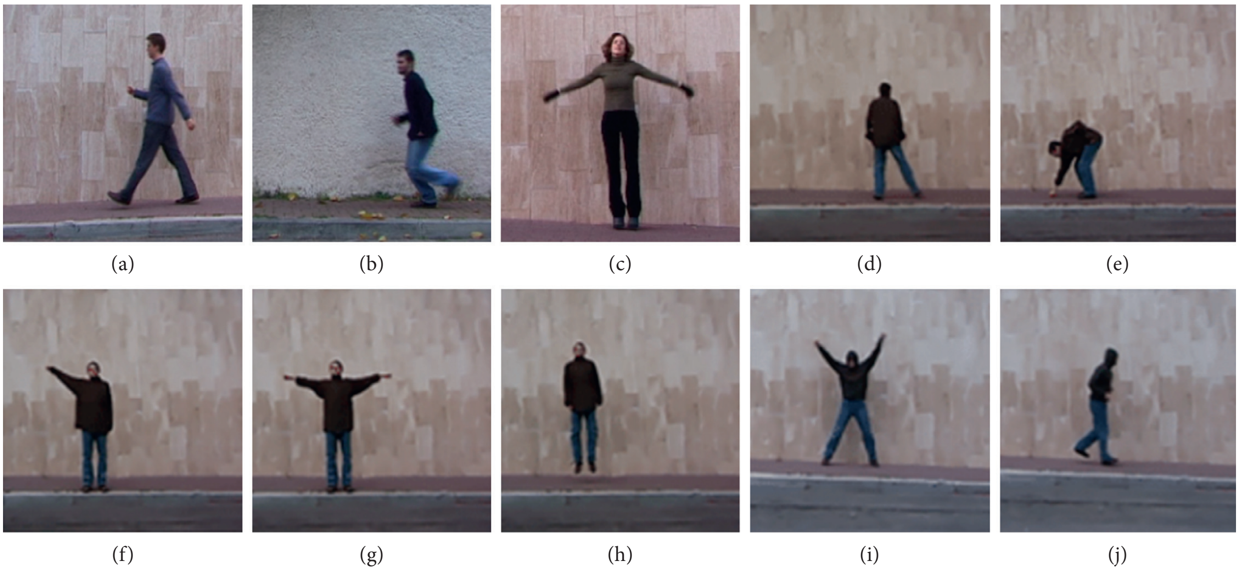
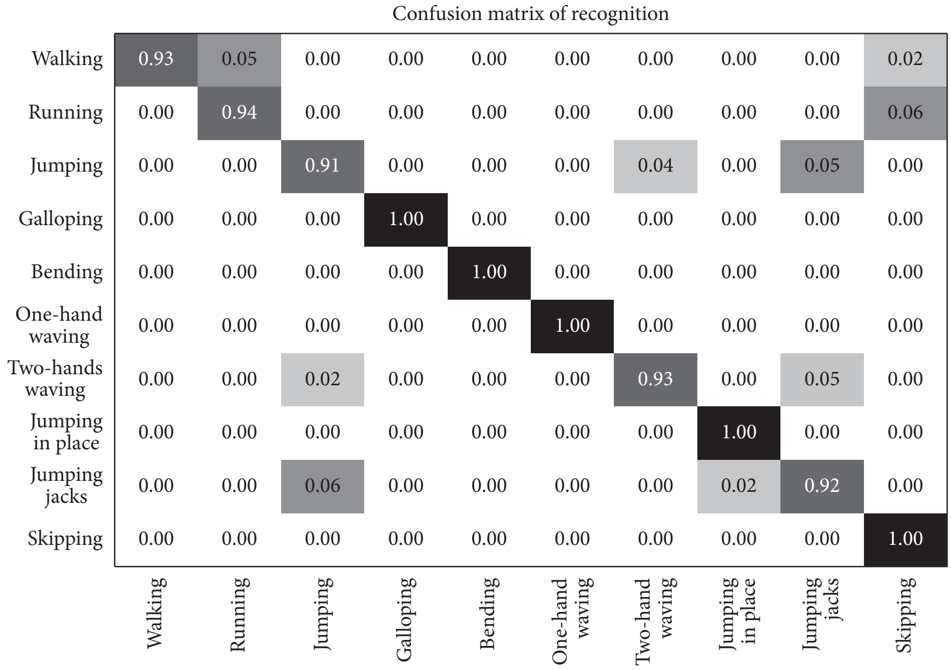


Figure 3: A Weizmann dataset with the actions considered for the proposed approach: (a) walking; (b) running; (c) jumping; (d) galloping; (e) bending; (f) one-hand waving; (g) two-hand waving; (h) jumping in place; (i) jumping jacks; (j) skipping.


Confusion matrix of recognition

	Walking	Running	Jumping	Galloping	Bending	One-hand waving	Two-hand waving	Jumping in place	Jumping jacks	Skipping
Walking	0.93	0.05	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02
Running	0.00	0.94	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.06
Jumping	0.00	0.00	0.91	0.00	0.00	0.00	0.04	0.00	0.05	0.00
Galloping	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
Bending	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00
One-hand waving	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00
Two-hands waving	0.00	0.00	0.02	0.00	0.00	0.00	0.93	0.00	0.05	0.00
Jumping in place	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00
Jumping jacks	0.00	0.00	0.06	0.00	0.00	0.00	0.00	0.02	0.92	0.00
Skipping	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00

Figure 4: Confusion matrix for the Weizmann dataset.

patches. Moreover, it was observed that the average recognition of the model decreases based on the influence of the number of output neurons.

5.2. CAVIAR Dataset. The context-aware vision using image-based active recognition (CAVIAR) is a video dataset [100]. The dataset consists of seven activities such as walking, slumping, fighting, entering, exiting, browsing, and meeting, as shown in Figure 6. The video sequences were recorded at different locations using a wide-angle camera lens in the INRIA Labs located in France and at a shopping center in Lisbon. The ground truth file is available in the CVML format. The file contains two types of labeling: activity label and scenario label. For every individual, the tracked target comprises 17 sequences and the pixel positions depend on image scaling. The second video sequence displays the frontal view and is synchronized frame by frame. The sequences are 1500 frames longer than the first sequence. The France sequence is categorized as “d1,” and the Lisbon sequence is classified as “d2.”

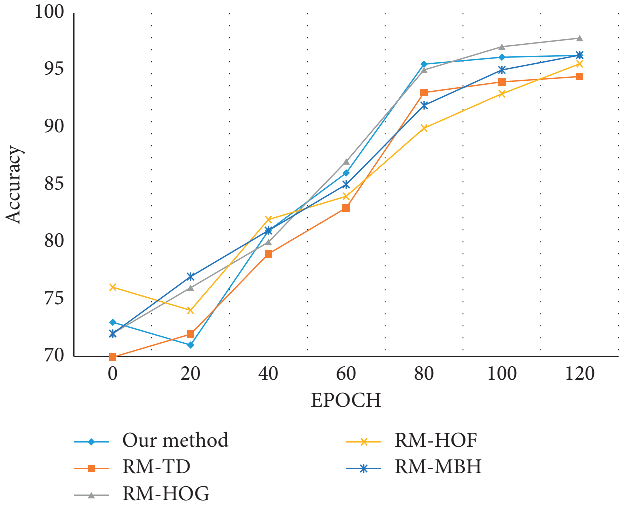


Figure 5: Evaluation of the classification accuracy of the Weizmann dataset.

Table 3: Average recognition rate for Weizmann dataset.

Method type	Average recognition rate (%)
Reference method using TD [110]	94.44
Reference method using HOG [110]	97.77
Reference method using HOF [110]	96.66
Reference method using MBH [110]	95.55
Reference method using combined methods [111]	96.66
Proposed method	96.3

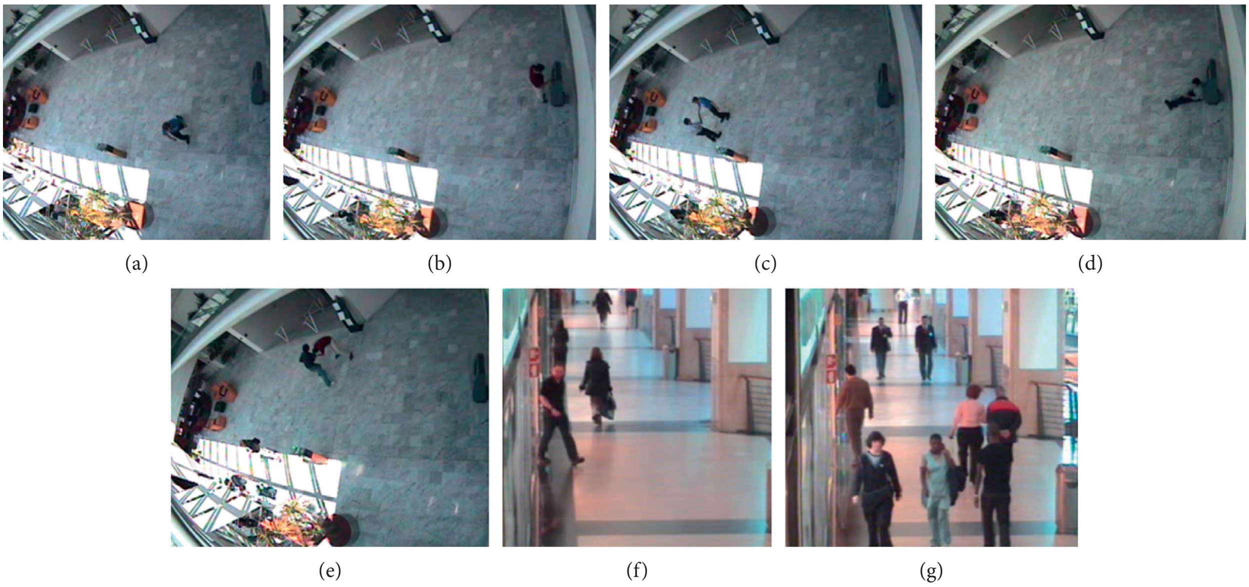


Figure 6: CAVIAR dataset with the sample actions: (a) walking; (b) browsing; (c) meeting; (d) slumping; (e) fighting; (f) exiting; (g) entering.

The size parameter is set to $N = 100$ and the effectiveness of the recognition method involved classifying two datasets. For the SVM classifier, the penalty coefficient was fixed as $P = 10$ and other slack variables are fixed by adaptive matching. The training set was categorized into 20 actions for the validation set and 9 actions for the training set. From the confusion matrix shown in Figure 7, it can be seen that some confusions are observed for the actions walking, entering, and exiting. Moreover, similarities were also observed for the actions of fighting and meeting. The other actions in the dataset are classified accurately.

The proposed method was compared with the reference method [112] implemented using the MFS detector and OpenCV classifier.

The results from Table 4 and Figure 8 show that the recognition rate from our proposed method for both labels "d1" and "d2" is significantly better than the reference method. Negri et al. [112] proposed an approach for pedestrian detection using movement feature space (MFS) to detect the movements and descriptor generation using a cascade of boosted classifiers. The validation of the MFS detector is performed using an SVM classifier. The reference method considered only the frontal view of the dataset resulting in only a few samples used for validation purposes. The less recognition rate achieved by the OpenCV detector 20 (20 stages) and OpenCV detector 25 (25 stages) because both classifiers require more stages for training to reduce the occurrence of false detection.

5.3. UCF Sports Action Dataset. The UCF sports human dataset [101] is comprised of 150 videos with 10 action categories. The ten categories of actions include walking, kicking, lifting, golfing, running, diving-side, horse-driving, swing-side angle, skateboarding, and bench swinging, as shown in Figure 9. The 150 video samples are divided into 102 samples for the training set and 48 samples for the testing set.

The N parameter for each cell is set to 200, and the penalty coefficient is set to $P = 10$ along with slack variables. The confusion matrix shown in Figure 10 shows a perfect accuracy rate with confusion observed only in the activities running and skateboarding as the model displayed false classification between these two action categories.

The recognition rate for the reference methods [113–116] is specified in Table 5.

Mironică et al. [113] proposed an approach to combine the frame features to model a global descriptor. The recognition accuracy of this method is affected when all the features are aggregated within a single descriptor and the BoW representation. Le et al. [114] proposed an unsupervised feature learning technique to learn the features directly from the video. They also explore an extended version of the ISA algorithm for learning the spatial-temporal features from the unlabeled data. The classification was performed using a multiclass SVM where the labels are predicted for all clips except the flipped versions resulting in a drop in accuracy.

An action region proposal method was provided by Rezazadegan et al. [115] using optical flows. Action detection and recognition were performed using CNN based on pose appearance and motion. Souly et al. [116] proposed an unsupervised method for detection using visual saliency [117] in videos. The video frames are divided into nonoverlapping cuboids and segmented using hierarchical segmentation to obtain the supervoxels from the cuboids. The features are decomposed into sparse matrices using PCA. When compared with the reference methods, the proposed method shows a better accuracy rate, as shown in Figure 11.

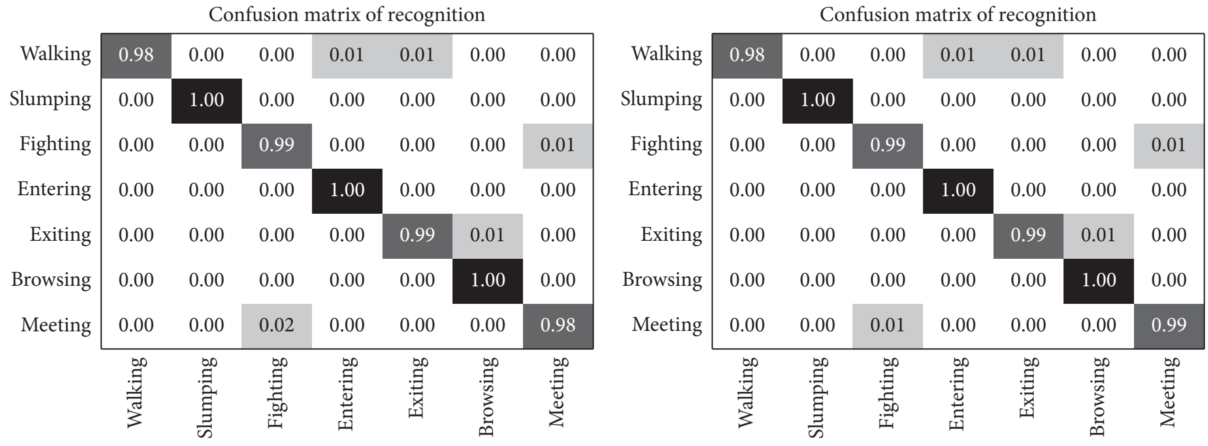
5.4. KTH Action Dataset. The KTH action dataset [102] is collated by the KTH Royal Institute of Technology. It is a video database that is comprised of human actions captured in various scenarios. It consists of six actions that include walking, boxing, running, waving, jogging, and clapping. The dataset is comprised of 600 video files that are a combination of 25 individuals, 6 actions, and 4 different types of scenarios, as shown in Figure 12.

The experimental analysis is carried out using the reference methods [118–122]. Only one-third of the video samples are considered for experimentation. The 200 video samples are divided into 140 samples for the training set and 60 samples for the testing set. The confusion matrix for the dataset is shown in Figure 13. It can be observed that the classification rate was affected by the action category running, as it was detected as walking. The action category jogging was classified as running.

During experimentation, the parameter is fixed as $N = 300$ with four scenarios labeled as "d1," "d2," "d3," and "d4." The penalty coefficient is set as $N = 10$, and the slack variables are obtained by adaptive data matching. The average recognition rate for the dataset is shown in Table 6.

Sreeraj et al. [118] proposed a multiposture human detection system based on HOG and BO descriptors. This approach shows a slightly better accuracy rate as the system uses a fast-additive SVM classifier. This combined approach retains the HOG precision rate to improve the detection rate. Yang et al. [119] constructed a neighborhood by adding weights on the distance components. SONFs and MONFs are generated by concatenating multiple SONFs. The method also uses LGSR classifier for obtaining the multiscale-oriented features and achieves better classification. Ji et al. [120] proposed an improved interest point detection to extract the 3D SIFT descriptors from single and multiple frames by applying PCA. The quantification of combined features using SVM increases computational cost and causes a drop in accuracy rate. STLPC descriptor was proposed by Shao et al. [121] and learns the spatial-temporal features from the video sequence. A Laplacian pyramid is constructed by maxpooling to capture the structural and motion features efficiently. The proposed method shows a slight decrease in 0.11% and 1.4%. The classification accuracy for the KTH dataset is shown in Figure 14.

5.5. CASIA Action Dataset. The CASIA dataset [103] is comprised of 8 human actions such as running, walking, jumping, crouching, punching, wandering, bending, and falling. The video action sequences were captured using a static camera from various angles and views. There are 1446 video sequences performed by 24 different subjects, as



	Walking	Slumping	Fighting	Entering	Exiting	Browsing	Meeting
Walking	0.98	0.00	0.00	0.01	0.01	0.00	0.00
Slumping	0.00	1.00	0.00	0.00	0.00	0.00	0.00
Fighting	0.00	0.00	0.99	0.00	0.00	0.00	0.01
Entering	0.00	0.00	0.00	1.00	0.00	0.00	0.00
Exiting	0.00	0.00	0.00	0.00	0.99	0.01	0.00
Browsing	0.00	0.00	0.00	0.00	0.00	1.00	0.00
Meeting	0.00	0.00	0.02	0.00	0.00	0.00	0.98

(a)

	Walking	Slumping	Fighting	Entering	Exiting	Browsing	Meeting
Walking	0.98	0.00	0.00	0.01	0.01	0.00	0.00
Slumping	0.00	1.00	0.00	0.00	0.00	0.00	0.00
Fighting	0.00	0.00	0.99	0.00	0.00	0.00	0.01
Entering	0.00	0.00	0.00	1.00	0.00	0.00	0.00
Exiting	0.00	0.00	0.00	0.00	0.99	0.01	0.00
Browsing	0.00	0.00	0.00	0.00	0.00	1.00	0.00
Meeting	0.00	0.00	0.01	0.00	0.00	0.00	0.99

(b)

Figure 7: Confusion matrix for the CAVIAR dataset with labels (a) d1 and (b) d2.

Table 4: Average recognition rate for CAVIAR dataset.

Method type	Average recognition rate(%)
Reference with MFS detector [112]	81.82
Reference with OpenCV detector 20 [112]	79.84
Reference with OpenCV detector 25 [112]	73.61
Proposed method d1	99.14
Proposed method d2	99.28

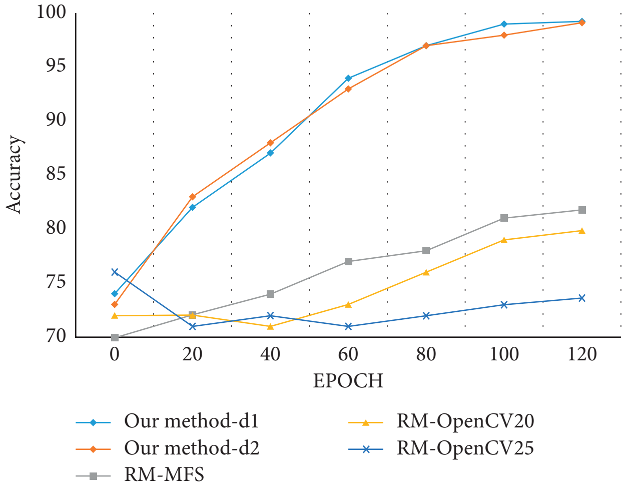


Figure 8: Evaluation of the classification accuracy of the CAVIAR dataset.

shown in Figure 15. For the experimental analysis, 250 video sequences are analyzed. They are split into 190 samples for the training set and 60 samples for the testing set. The N parameter is set as 300 for every cell, while the penalty coefficient is set as $P = 10$ along with the respective slack variables. The reference framework [123] using the EM technique using an M-class SMV classifier and other classifiers is provided in Table 7.

The confusion matrix in Figure 16 shows that the action category falling achieves a full accuracy rate. Similar action categories such as running, walking, crouching, and bending have a 99% accuracy rate. The categories of punching and wandering show the least accuracy rate of 98%.

Table 7 shows the average recognition rate for the CASIA dataset. Sharif et al. [123] proposed a hybrid strategy for human action classified by the integration of four major techniques. Initially, the objects in motion are uniformly segmented, and the features are extracted using LBP, HOG, and Haralick features. The feature selection is performed by the joint entropy-PCA method, and the classification is performed using multiclass SVM. The following classifiers multiclass SVM, DT, LDA, KNN, and EBT are used for experimental analysis. If high-resolution videos are used, there is a drop in efficiency due to computation overhead.

Figure 17 shows that our proposed method has a better recognition rate when compared to the classifier used in the reference method.

5.6. i3DPost Multiview Dataset. The i3DPost dataset is a multiview/3D human action/interaction database [104] created by the University of Surrey and CERTH-ITI (Center of Research and Technology Hellas Informatics and Telematics Institute). The dataset consists of multiview videos and 3D posture model sequences. The videos were recorded using the convergent eight-camera setup for capturing high-definition images with twelve people performing twelve different types of human motions. The actions performed by the subjects include walking, running, bending, jumping, waving, handshaking, pulling, and facial expressions, as shown in Figure 18. The 104 video sequences are divided into 60 samples for the training set and 44 samples for the testing set. This is because the action in this dataset is much more complex than the UCF sports action dataset. The N parameter is set as 150 for every cell, while the penalty coefficient is set as $P = 10$ along with the respective slack variables.

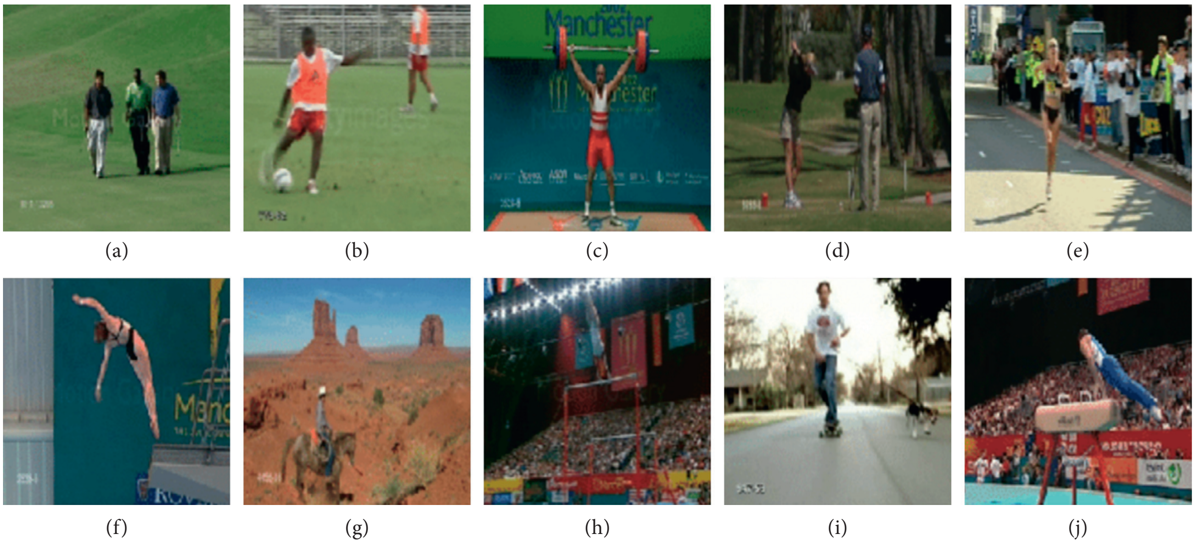
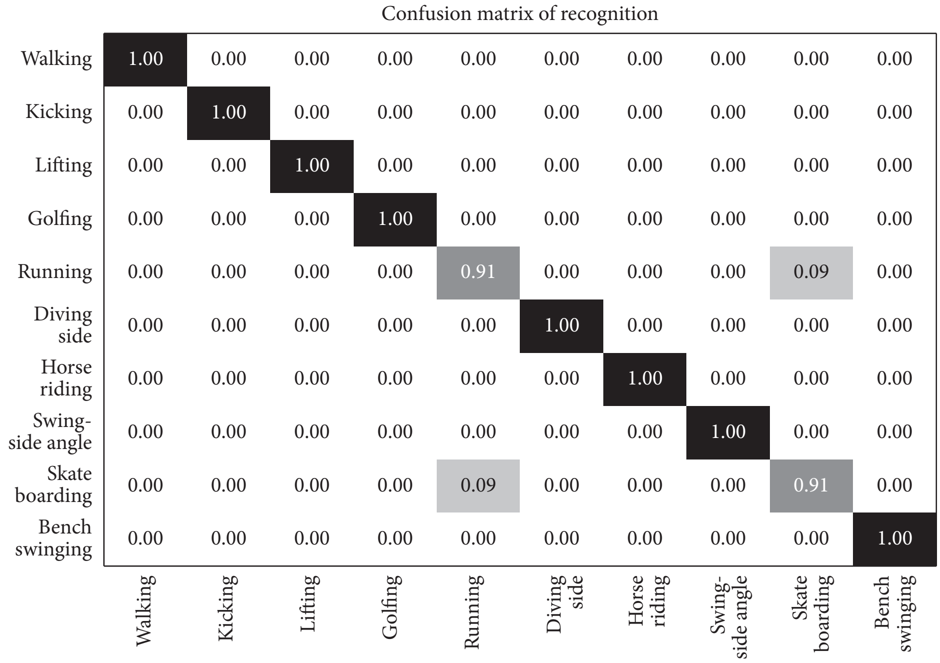


FIGURE 9: Actions considered for the proposed approach in the UCF sports action dataset: (a) walking; (b) kicking; (c) lifting; (d) golfing; (e) running; (f) diving side; (g) horse riding; (h) swing-side angle; (i) skate boarding; (j) bench swinging.



	Walking	Kicking	Lifting	Golfing	Running	Diving side	Horse riding	Swing-side angle	Skate boarding	Bench swinging
Walking	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Kicking	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Lifting	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Golfing	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
Running	0.00	0.00	0.00	0.00	0.91	0.00	0.00	0.00	0.09	0.00
Diving side	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00
Horse riding	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00
Swing-side angle	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00
Skate boarding	0.00	0.00	0.00	0.00	0.09	0.00	0.00	0.00	0.91	0.00
Bench swinging	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00

FIGURE 10: Confusion matrix for the UCF sports action dataset.

TABLE 5: Average recognition rate for UCF sports action dataset.

Method type	Average recognition rate (%)
Rezazadegan et al. [115]	93.3
Mironică et al. [113]	74.1
Souly et al. [116]	88.6
Le et al. [114]	86.8
Proposed method	98.2

The confusion matrix obtained in Figure 19 shows that action categories jumping, bending, waving, stand-up, run-fall, and walk-sit have a full recognition rate. The actions running and walking have a misclassification rate in a few scenarios. Also, the actions handshaking and pulling are misclassified due to similar poses in some frames leading to a decrease in recognition rate.

In Table 8, Gkalelis et al. [124] and Iosifidis et al. [125] proposed an approach using binary masks obtained from multiview posture images for vectorization. This technique was used to extract the low-dimensional feature descriptors. DFT, FVQ, and LDA are applied for action recognition and classification. The authors tested their method with a limited testing set comprising only eight actions

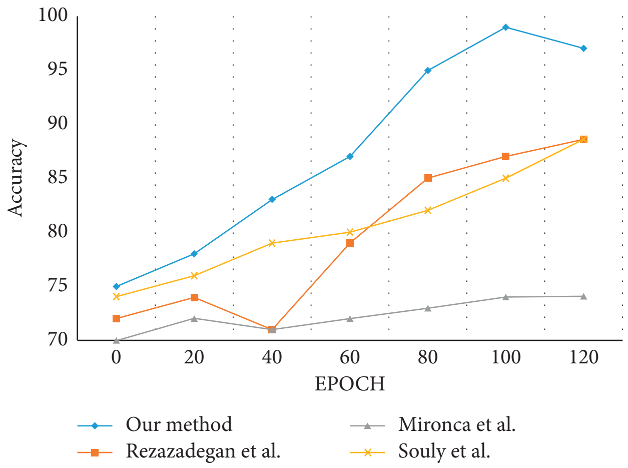
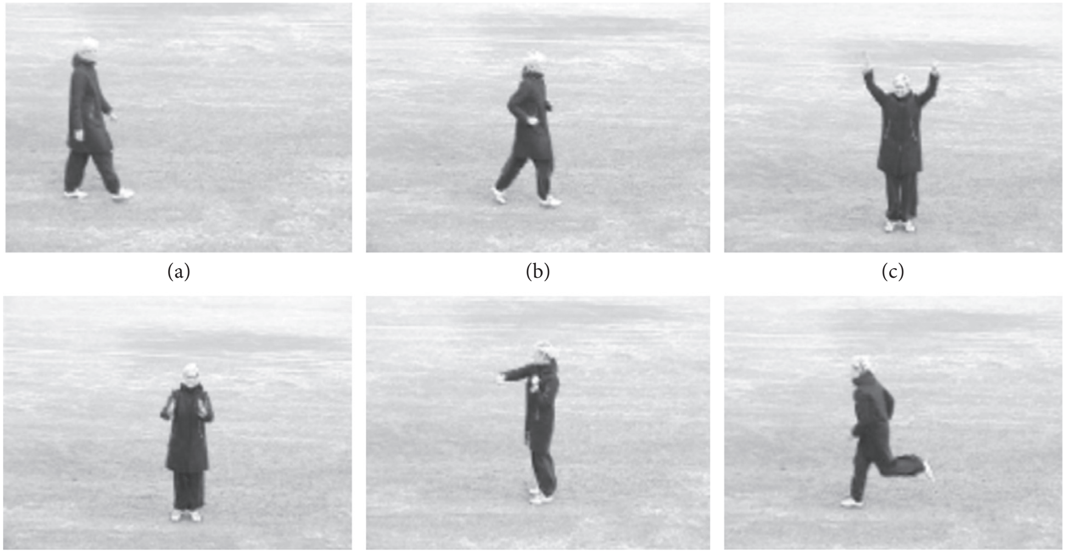


Figure 11: Evaluation of the classification accuracy for UCF sports action dataset.



(a)

(b)

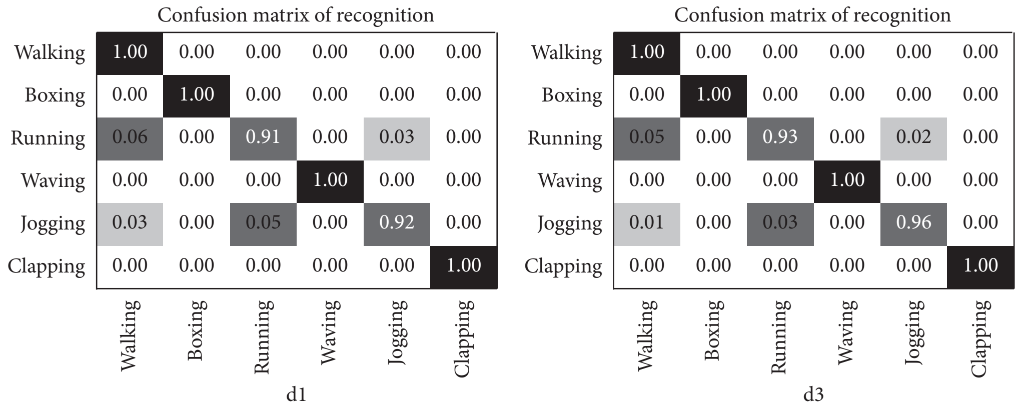
(c)

(d)

(e)

(f)

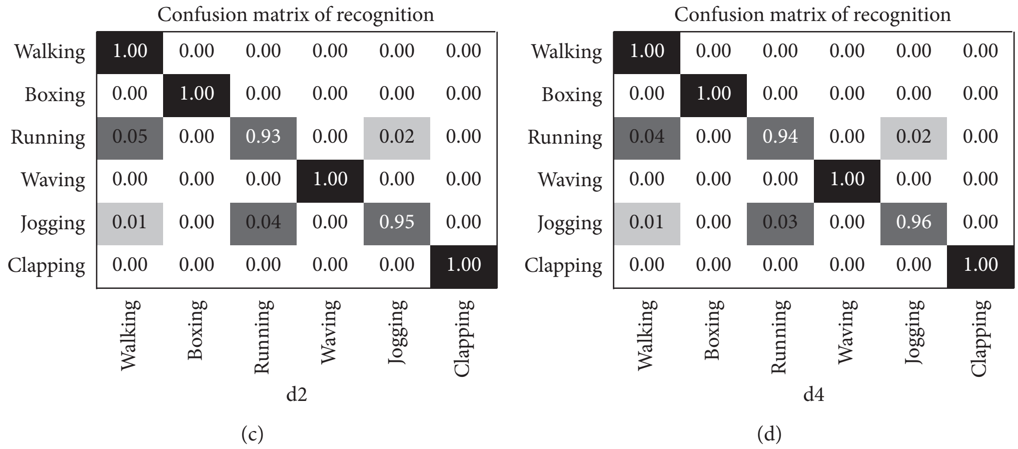
Figure 12: Actions considered for the proposed approach in the KTH dataset: (a) walking; (b) jogging; (c) waving; (d) clapping; (e) boxing; (f) running.



(a)

(b)

Figure 13: Continued.



Confusion matrix of recognition

	Walking	Boxing	Running	Waving	Jogging	Clapping
Walking	1.00	0.00	0.00	0.00	0.00	0.00
Boxing	0.00	1.00	0.00	0.00	0.00	0.00
Running	0.05	0.00	0.93	0.00	0.02	0.00
Waving	0.00	0.00	0.00	1.00	0.00	0.00
Jogging	0.01	0.00	0.04	0.00	0.95	0.00
Clapping	0.00	0.00	0.00	0.00	0.00	1.00

d2

(c)

Confusion matrix of recognition

	Walking	Boxing	Running	Waving	Jogging	Clapping
Walking	1.00	0.00	0.00	0.00	0.00	0.00
Boxing	0.00	1.00	0.00	0.00	0.00	0.00
Running	0.04	0.00	0.94	0.00	0.02	0.00
Waving	0.00	0.00	0.00	1.00	0.00	0.00
Jogging	0.01	0.00	0.03	0.00	0.96	0.00
Clapping	0.00	0.00	0.00	0.00	0.00	1.00

d4

(d)

Figure 13: Confusion matrix for the KTH dataset with labels (a) d1, (b) d2, (c) d3, and (d) d4.

Table 6: Average recognition rate for KTH dataset.

Method type	Average recognition rate (%)
Sreeraj et al. [118]	95.21
Yang et al. [119]	96.50
Ji et al. [120]	94.92
Shao et al. [121]	95
Proposed method d1	94.5
Proposed method d2	92.9
Proposed method d3	94.8
Proposed method d4	95.1

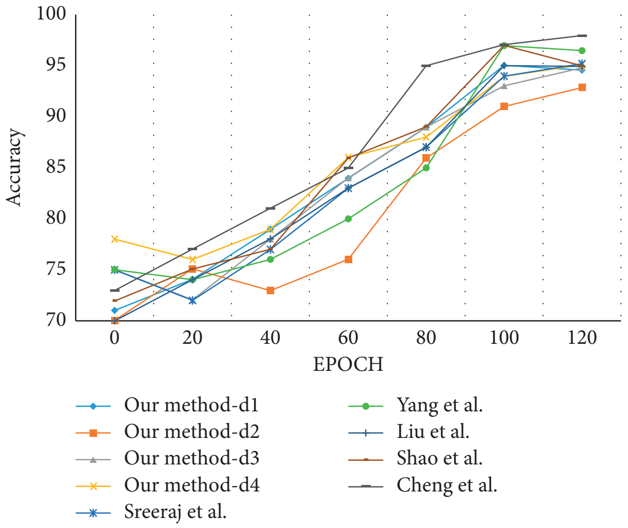


Figure 14: Evaluation of the classification accuracy for KTH dataset.

when compared to 13 actions used in our proposed approach.

Holte et al. [126] proposed a score-based fusion technique for extracting the spatial-temporal features. These feature vectors are efficient for high frame data capture with different densities and views. Based on the evaluation of the accuracy rate in Figure 20, the proposed method achieves significant performance when compared to other reference methods with 13 actions.

5.7. JHMDB Action Dataset. The joint-annotated human motion database [105] is categorized into 12 action types. The twelve actions shown in Figure 21 include walking, climbing, golfing, kicking, jumping, pushing, running, pull-up, catching, picking-up, baseball playing, and throwing.

The dataset comprises of three segmentation methods for the training and the testing set. For our experimentation, we are using only one segmentation method where only 316 videos are considered. They are further divided into 224 video segments for the training set and 92 video segments for the testing set. The N parameter is set as 350 for every cell, while the penalty coefficient is set as $P = 10$ along with the respective slack variables.

The confusion matrix from Figure 22 shows that the action categories climbing, golfing, kicking, pushing, pull-up, and pick-up have a 100 percent recognition rate. The action categories such as jumping, running, and catching showed recognition rates ranging from 91 to 98 percent. The action categories that showed the least performance were walking that was misclassified with running. The action jumping was misclassified as catching and vice versa, while the action baseball playing was misclassified as golfing.

From Table 9, Jhuang et al. [105] performed a systematic performance evaluation using the annotated dataset. The baseline model was evaluated by categorizing the poses in the sample into three categories: low-, middle-, and high-level features. The dataset is annotated using a 2D puppet model, and the optical flow or the puppet flow is computed. The low- and mid-level poses are evaluated using the dense trajectory technique, while the high-level poses are evaluated using NTraj. Yu et al. [127] proposed a multimodal three-stream network for action recognition. PoseConvNET is used for detecting the 2D poses using the 2D CMU pose estimator, and the interpolation method is introduced for joint completion. The analysis performed on the individual cues showed a less recognition rate when compared with the proposed method.

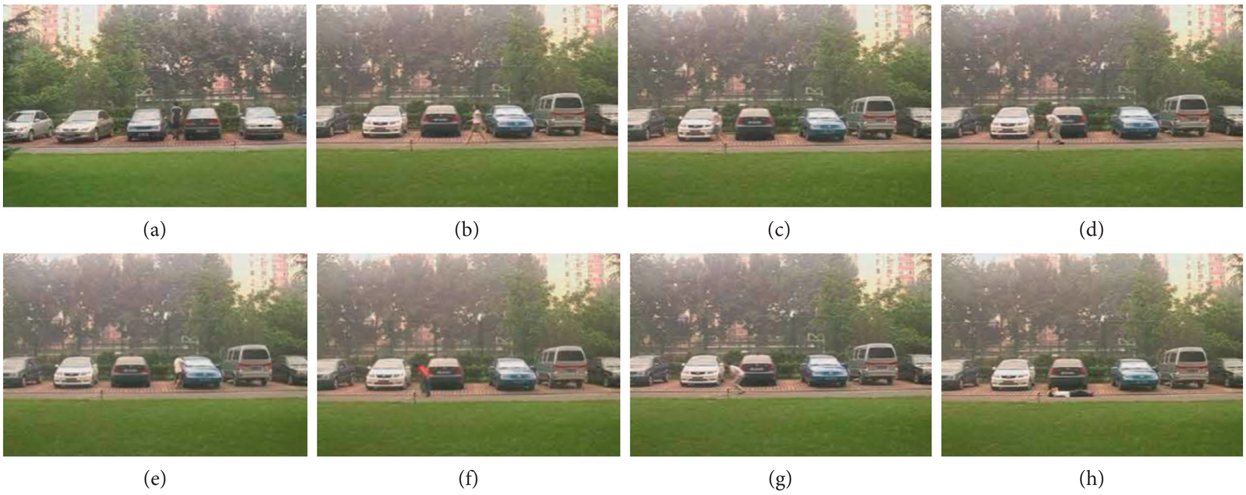
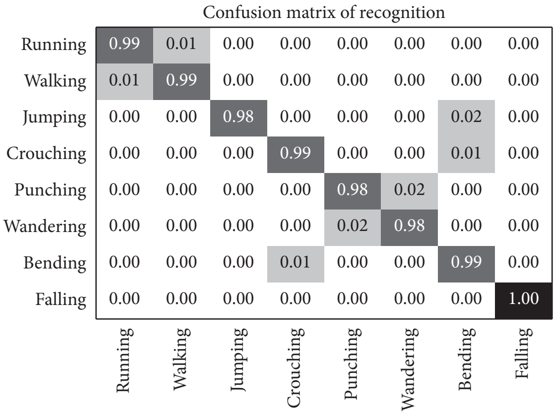


Figure 15: Action samples in the CASIA dataset: (a) wandering; (b) walking; (c) running; (d) crouching; (e) punching; (f) bending; (g) jumping; (h) falling.

Table 7: Average recognition rate for CASIA dataset.

Method type	Average recognition rate (%)
Reference framework with M-class SVM [123]	98.70
Reference framework with DT [123]	97.90
Reference framework with LDA [123]	98.20
Reference framework with KNN [123]	98.10
Reference framework with EBT [123]	98.10
Proposed method	98.75



Confusion matrix of recognition

	Running	Walking	Jumping	Crouching	Punching	Wandering	Bending	Falling
Running	0.99	0.01	0.00	0.00	0.00	0.00	0.00	0.00
Walking	0.01	0.99	0.00	0.00	0.00	0.00	0.00	0.00
Jumping	0.00	0.00	0.98	0.00	0.00	0.00	0.02	0.00
Crouching	0.00	0.00	0.00	0.99	0.00	0.00	0.01	0.00
Punching	0.00	0.00	0.00	0.00	0.98	0.02	0.00	0.00
Wandering	0.00	0.00	0.00	0.00	0.02	0.98	0.00	0.00
Bending	0.00	0.00	0.00	0.01	0.00	0.00	0.99	0.00
Falling	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00

Figure 16: Confusion matrix for the CASIA dataset.

However, when all the cues are combined, the reference method proposed by You et al. shows better recognition by 1.34 percent when compared to our proposed method. The evaluation of the accuracy rates for the model is shown in Figure 23.

5.8. UCF101 Action Dataset. The UCF101 [106] is a collection of human action dataset [128] and is an extended version of the UCF50 dataset. It is comprised of 101 human behaviors, and they are categorized into 25 groups, as shown in Figure 24. Every group is comprised of 13320 behavioral segment videos. The training and testing sets are divided into three categories. The average recognition rate from the three sets is analyzed from the dataset. The N parameter is set as 400 for every cell, while the penalty coefficient is set as $P = 10$, whereas other parameters are provided by pattern matching the image data to the processed image data.

The effectiveness of the algorithm is measured using the following reference algorithms [9, 93, 111, 129, 130], as shown in Table 10.

Ryoo [111] proposed a dynamic and integral BoW model for action prediction. The human activities are predicted using 3D spatial-temporal local features along with the interest points. The features values are clustered to form visual words using K-means and the Integral BoW used HOG descriptors. The method showed a drop in recognition rates during the early stages of detection. Cao et al. [129] proposed a probabilistic framework for action recognition. Sparse coding is applied to spatial-temporal features, and the likelihood is obtained using MSSC. The datasets were tested using SC and MSSC methods; the recognition rate was less satisfactory and required more training due to model complexity.

Kong et al. [130] proposed the MTSSVM model for predicting the temporal dynamics of all the observed features. This approach showed an improvement in the recognition rate when compared to other reference methods. The drop in recognition rate is because the model requires prior knowledge of the temporal action that can be achieved only via prolonged training. A mem-LSTM model was proposed by You et al. [9] for recording the hard samples. The model used CNN and LSTM on the partially observed videos. The model has an improved recognition rate as it does not require prior knowledge of the features, and the global memory is sufficient for prediction. From Figure 25, it can be observed that the proposed method outperforms all the other reference methods.

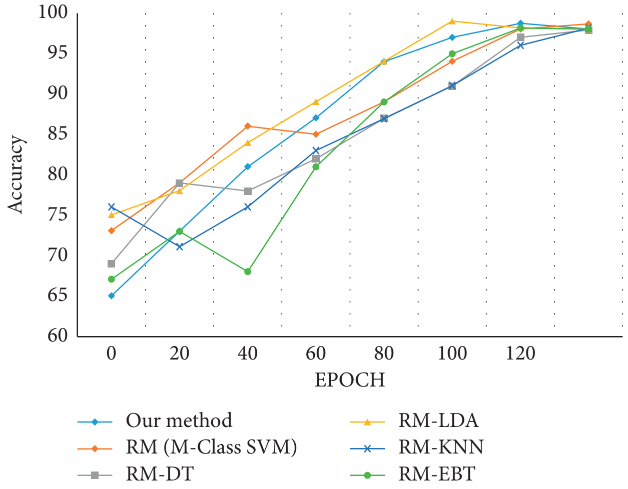


FIGURE 17: Evaluation of the classification accuracy for CASIA dataset.

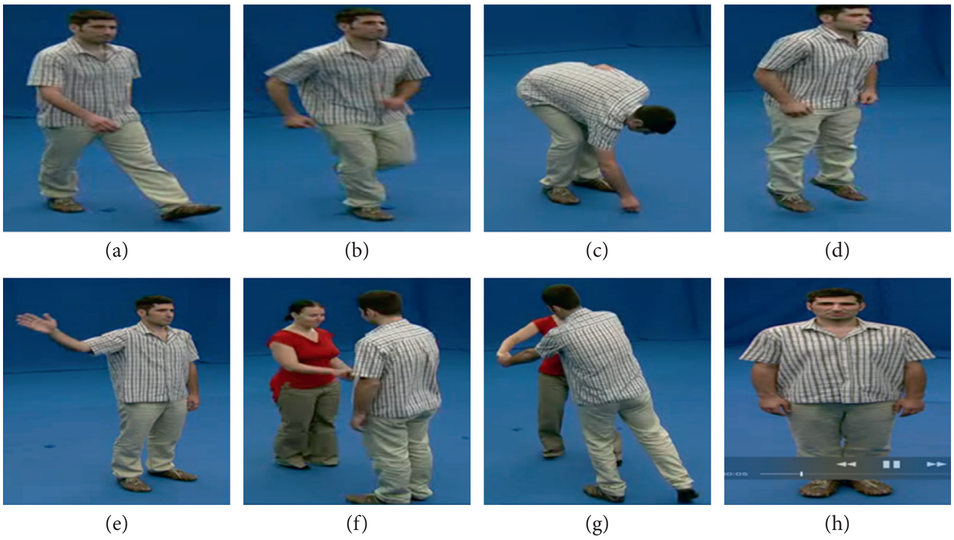


FIGURE 18: Actions considered in the i3DPost multiview dataset: (a) walking; (b) running; (c) bending; (d) jumping; (e) waving; (f) handshaking; (g) pulling; (h) face expressions.

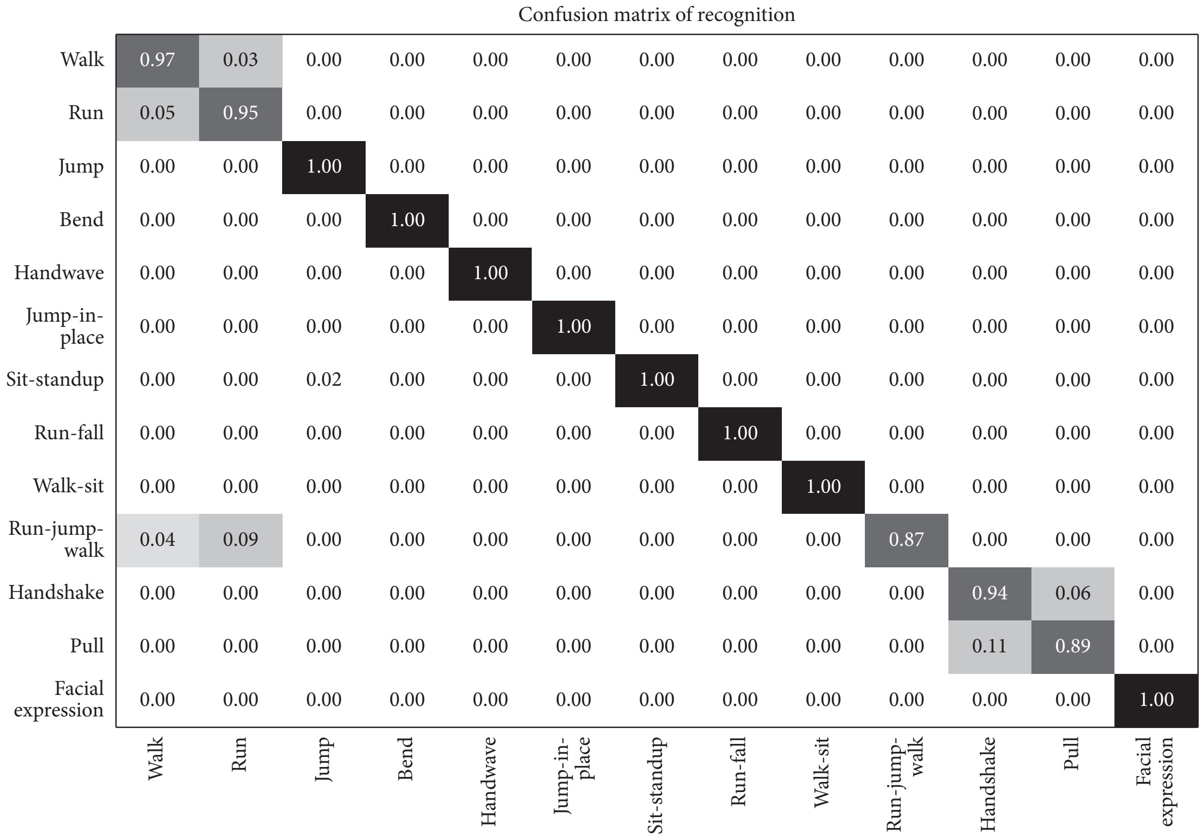
5.9. HMDB51 Action Dataset. The HMDB51 action dataset [107] is comprised of 51 behavior categories that contain 100 videos each and 6676 action sequences, as shown in Figure 26. The data are divided into three training and testing sequences for action recognition, 60 training videos, and 30 test videos. From Table 11, the proposed method is evaluated with other techniques. The N parameter is set as 150 for every cell, while the penalty coefficient is set as $P = 10$.

Jiang et al. [131] proposed a fuss-free method for modeling motion relationships by adopting the global and locale reference points. The code words are derived from the local feature patches and tested. Jain et al. [48] proposed a technique for decomposing the visual motion into dominant motions to compute the features and their respective trajectories. A DCS descriptor along with the VLAD coding technique is used for action recognition.

Heng et al. [132] introduced a technique for matching the feature points between the frames using the SURF descriptor and optical flow. These matched features are graphed with RANSAC for human action recognition. Zhang et al. [133] proposed a deep two-stream architecture for action recognition using video datasets. The knowledge is transferred from optical CNN to motion vector CNN to reduce computation overhead and to boost the performance of the model.

Karen et al. [135] proposed a two-stream ConvNet architecture to combine spatial-temporal features. The model is trained on dense multiframe optical flow to achieve enhanced performance. Figure 27 shows that the proposed method surpasses all the techniques considered for evaluation.

5.10. Influence of the **N** *Parameter, Model Accuracy, and Loss Function.* Restricted Boltzmann machine (RBM) is a



Confusion matrix of recognition

	Walk	Run	Jump	Bend	Handwave	Jump-in-place	Sit-standup	Run-fall	Walk-sit	Run-jump-walk	Handshake	Pull	Facial expression
Walk	0.97	0.03	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Run	0.05	0.95	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Jump	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Bend	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Handwave	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Jump-in-place	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Sit-standup	0.00	0.00	0.02	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
Run-fall	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00
Walk-sit	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00
Run-jump-walk	0.04	0.09	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.87	0.00	0.00	0.00
Handshake	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.94	0.06	0.00
Pull	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.11	0.89	0.00
Facial expression	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00

Figure 19: Confusion matrix i3DPost multiview dataset.

Table 8: Average recognition rate for i3DPost multiview dataset.

Method type	Average recognition rate (%)
Gkalelis at al. [124] (with 8 actions)	90.00
Iosifidis et al. [125] (with 8 actions)	90.88
Holte et al. [126] (with 8 actions)	92.19
Proposed method (with 13 actions)	97.07

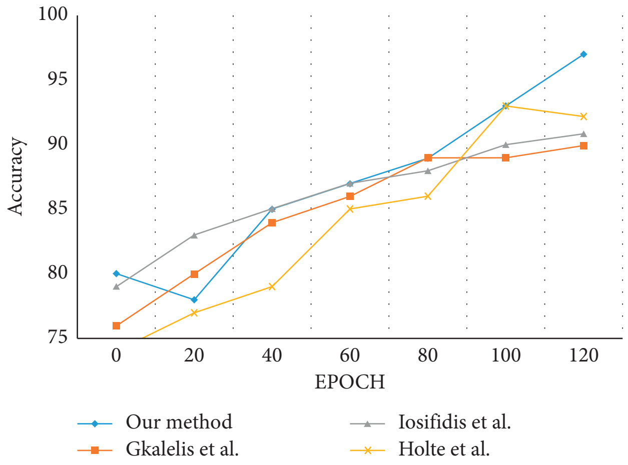


Figure 20: Evaluation of the classification accuracy for i3DPost multiview dataset.

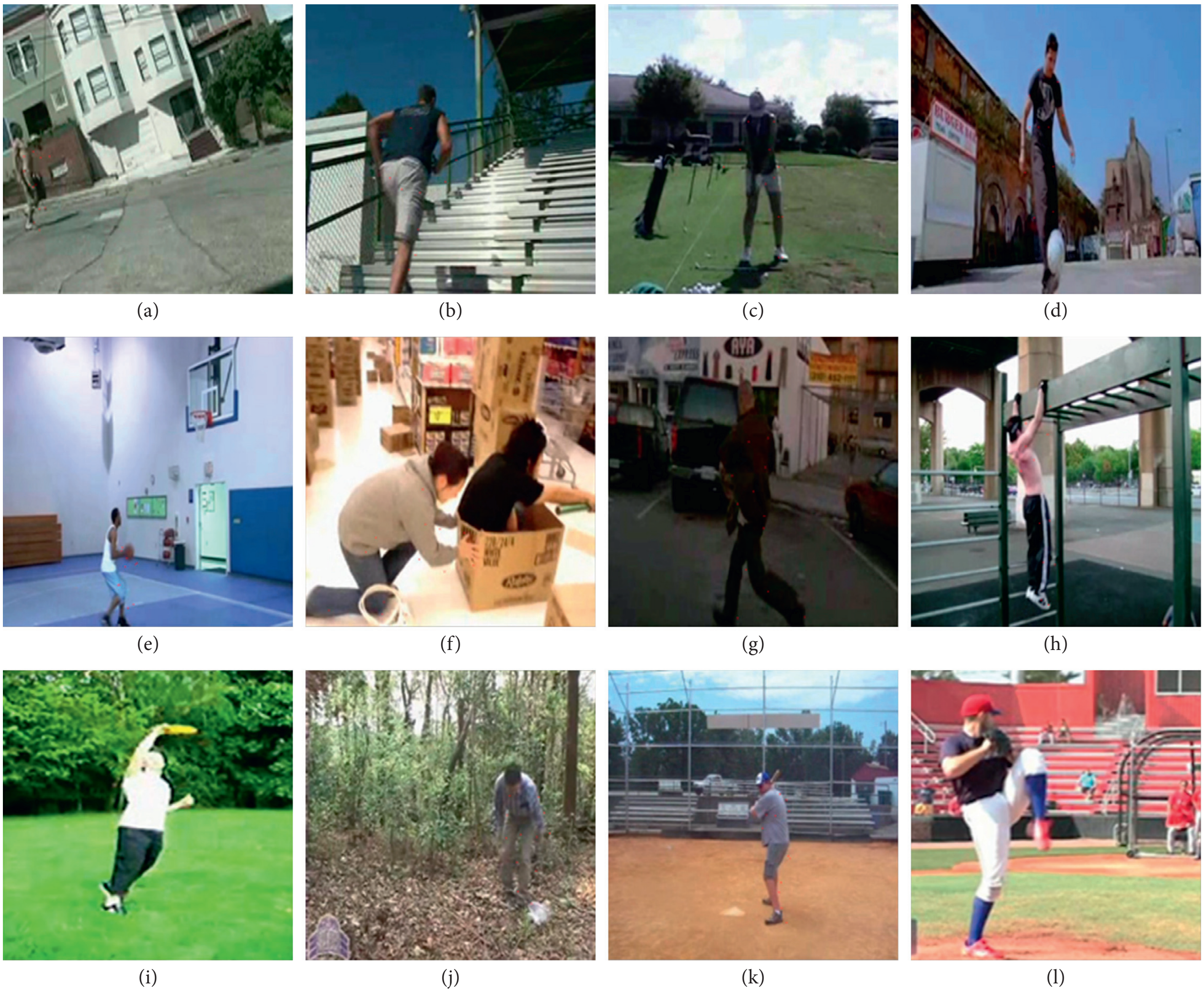
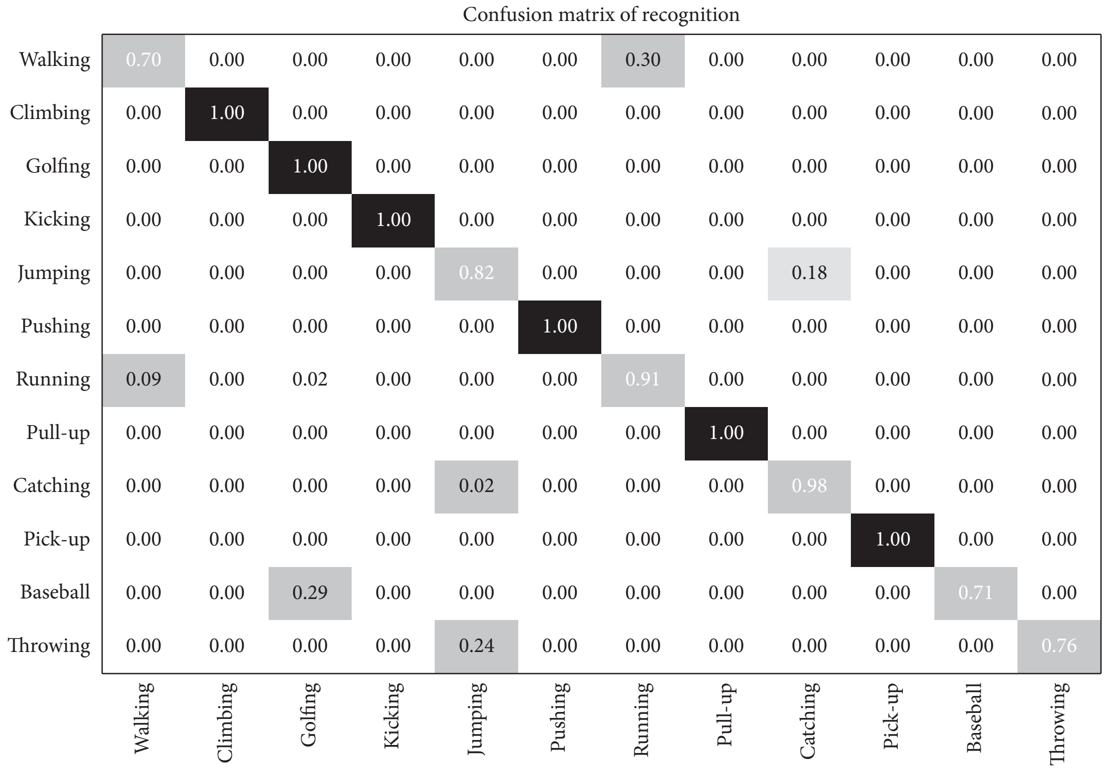


Figure 21: Actions considered for training in the JHMDB action dataset: (a) walking; (b) climbing; (c) golfing; (d) kicking; (e) jumping; (f) pushing; (g) running; (h) pull up; (i) catching; (j) picking up; (k) baseball playing; (l) throwing.

stochastic autoencoder that functions as both encoder and decoder. It is used for weight initialization in a neural network before training using stochastic gradient descent (SDG) for backpropagation. During training, multiple RBMs are stacked on top of each other to form a neural network. The RBM layer in the neural network inherits the functionality of the network. Thus, it can function as both an autoencoder or as a part of the neural network. As mentioned earlier, the RBM-NN comprises a two-layer neural network that is fully connected to other layers. The visible layer functions as the input layer, and the hidden layer corresponds to the features of the input neurons. During training, the RBMs adjust their weights automatically. The weight fed to one output neuron corresponds to one feature of the input. For instance, each weight originates from an input pixel, and the value determines the strength of the connection towards the activation function. The parameters generated by RBM are dynamic, and minor changes can cause huge differences in network behavior and performance. Every neuron is assigned to an activation function, and the node output is either set as 1 (on) or 0 (off).

From Figure 28, we can observe that the classification accuracy of the model is influenced by the number of neurons provided to the RBM. The classification rate reaches the highest when it satisfies the N parameter and gradually decreases after crossing the threshold layer. The influence of the parameter for the all the datasets shows similar results.

Deep learning neural networks are trained using the SDG optimization algorithm. As a part of the optimization problem, it is essential to evaluate the error rate for the current state of the model continuously. The error function used for our proposed method is a logistic regression loss function that estimates the loss of the models for weight updation. The loss function for our model is evaluated by generating a regression problem with a set of input variables, noise, and other properties. For evaluation, 100 input features are defined as input to the model. A total of 1000 samples will be randomly generated, and the pseudorandom number generator is fixed to 1 to ensure that the same number of samples is considered every time the model is evaluated. Each input and the output variable follows Gaussian distribution for data standardization. The model



Confusion matrix of recognition

	Walking	Climbing	Golfing	Kicking	Jumping	Pushing	Running	Pull-up	Catching	Pick-up	Baseball	Throwing
Walking	0.70	0.00	0.00	0.00	0.00	0.00	0.30	0.00	0.00	0.00	0.00	0.00
Climbing	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Golfing	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Kicking	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Jumping	0.00	0.00	0.00	0.00	0.82	0.00	0.00	0.00	0.18	0.00	0.00	0.00
Pushing	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
Running	0.09	0.00	0.02	0.00	0.00	0.00	0.91	0.00	0.00	0.00	0.00	0.00
Pull-up	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00
Catching	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.98	0.00	0.00	0.00
Pick-up	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00
Baseball	0.00	0.00	0.29	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.71	0.00
Throwing	0.00	0.00	0.00	0.00	0.24	0.00	0.00	0.00	0.00	0.00	0.00	0.76

Figure 22: Confusion matrix for JHMDB action dataset.

Table 9: Average recognition rate for JHMDB action dataset.

Method type	Average recognition rate (%)
Reference baseline model [105]	56.6
Reference baseline with low/mid-level pose [105]	69.00
Reference baseline with high-level pose [105]	76.00
Reference method with RGB + flow [127]	95.04
Reference method with RGB + pose [127]	91.67
Reference method with flow + pose [127]	97.10
Reference method with all combinations [127]	98.41
Proposed method	97.07

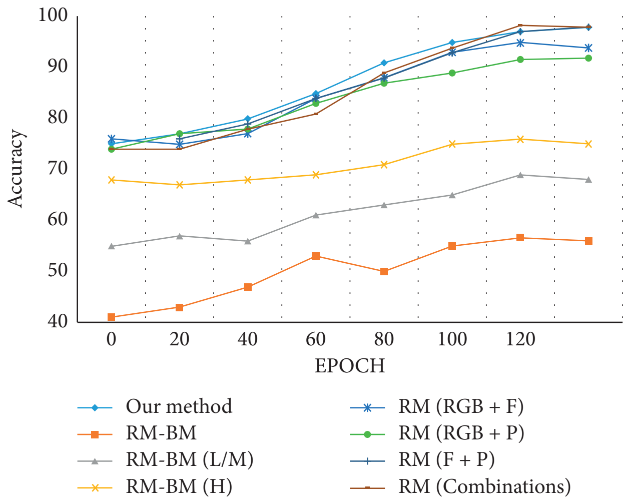


Figure 23: Evaluation of the classification accuracy for JHMDB action dataset.

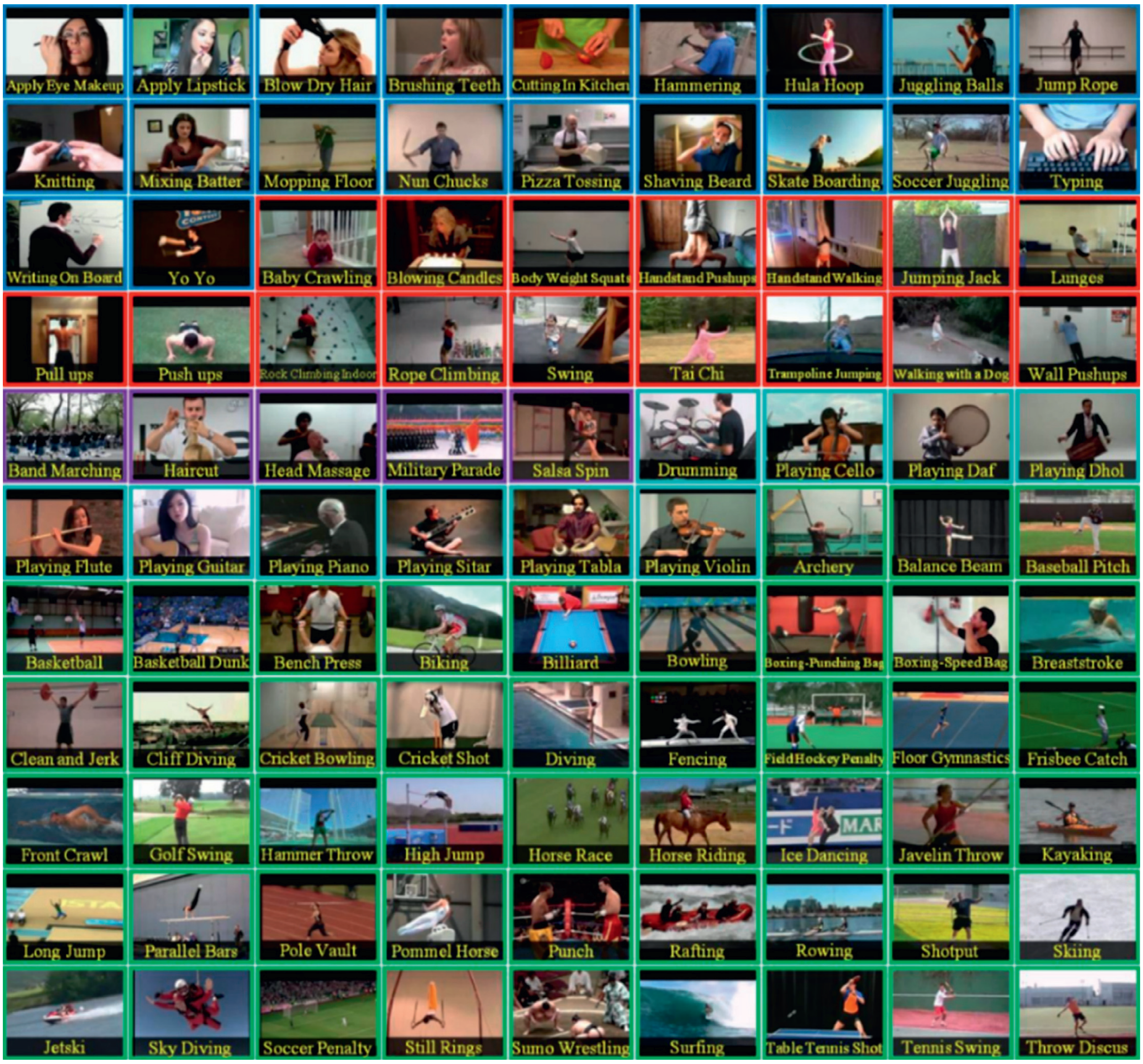


Figure 24: Action samples in the UCF101 action dataset.

Table 10: Average recognition rate for UCF101 action dataset.

Method type	Average recognition rate (%)
Dynamic BoW [111]	53.16
Integral BoW [111]	74.39
MSSC [129]	61.79
MTSSVM [130]	82.39
DeepSCN [93]	85.75
Mem-LSTM [9]	88.37
Proposed method	88.64

has the learning rate set to 0.1 with learning momentum set to 0.9. The model is trained for 100 epochs, and the testing set is evaluated at the end of every epoch to compute the loss function for the model. Figure 29 shows the performance of the model for the training and testing sets. Since the input and target variable for the model follow Gaussian distribution, the average of the squared differences between the actual and predicted values are computed.

If the difference is large, a strict penalty is enforced on the model for making a misclassification. From Figure 30(a), we observe that model was capable of learning the problem by achieving near-zero error for MSE loss. The model converges reasonably for the training and the testing set with a good performance rate.

In case, if the target value consists of widespread values or the difference is large, punishing the model by enforcing a

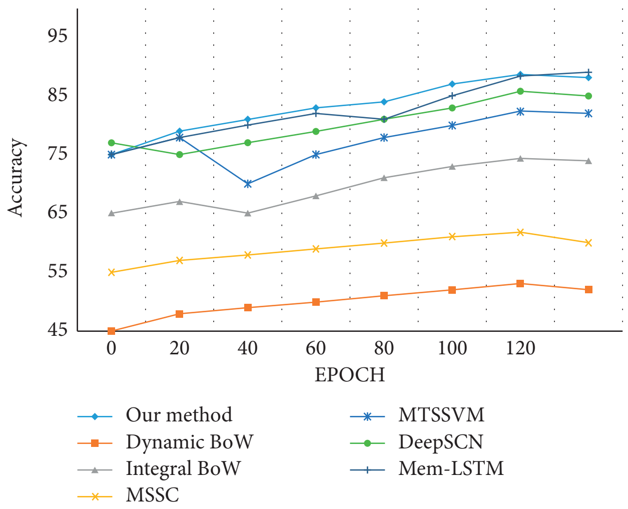


Figure 25: Evaluation of the classification accuracy for JHMDB action dataset.

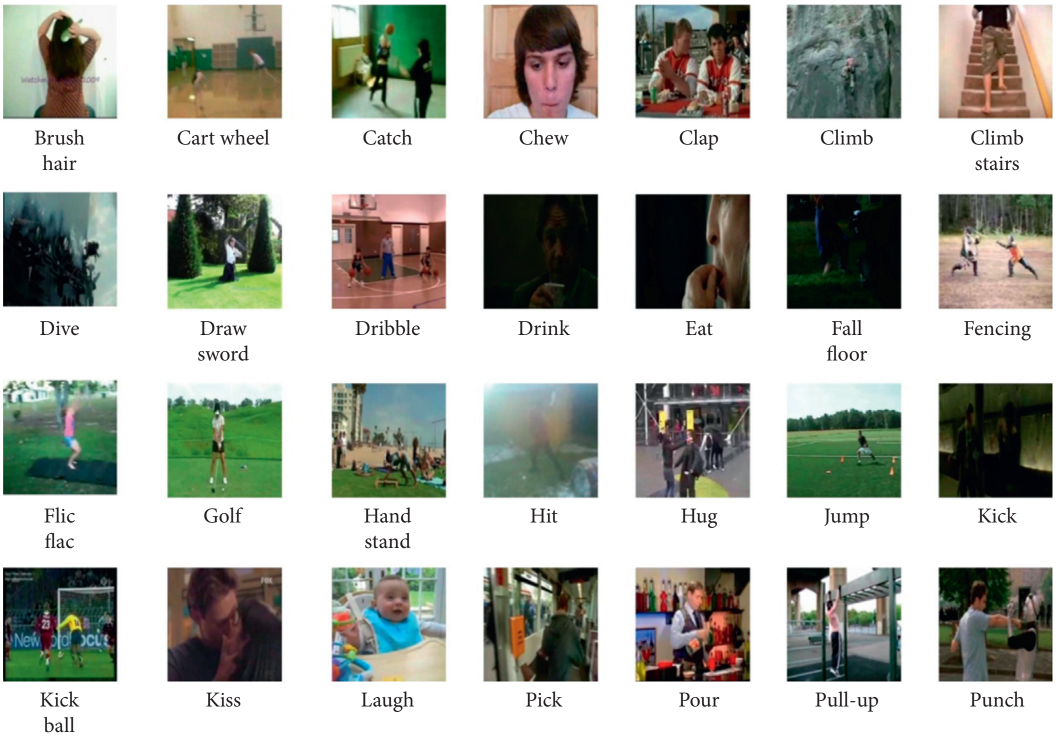


Figure 26: Action samples in the HMDB51 action dataset.

Table 11: Average recognition rate for the HMDB51 dataset.

Method type	Average recognition rate (%)
Jiang et al. [131]	40.7
Jain et al. [48]	52.1
Heng et al. [132] without HD	55.9
Heng et al. [132] with HD	57.2
Zhang et al. [133]	50.6
Wang et al. [134]	46.7
Karen et al. [135]	59.4
Proposed method	59.21

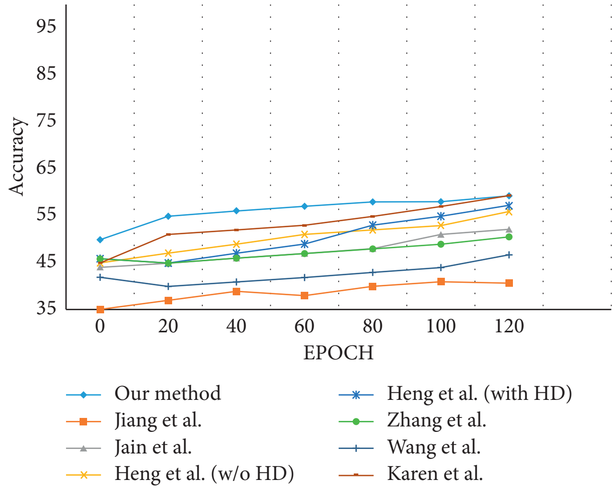


FIGURE 27: Evaluation of the classification accuracy for HMDB51 action dataset.

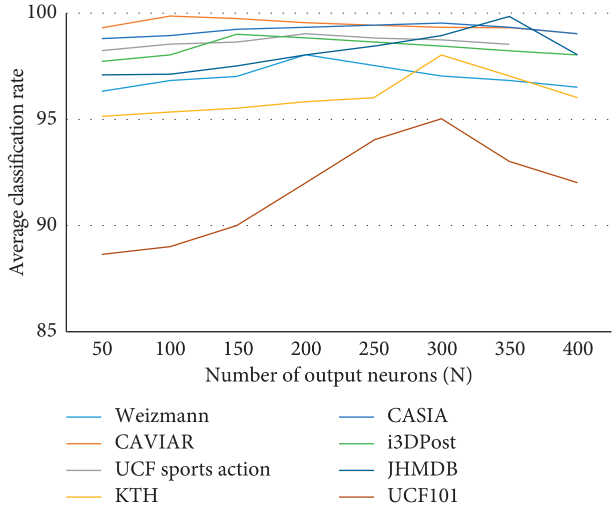


FIGURE 28: Influence of N parameter on the classification rate.

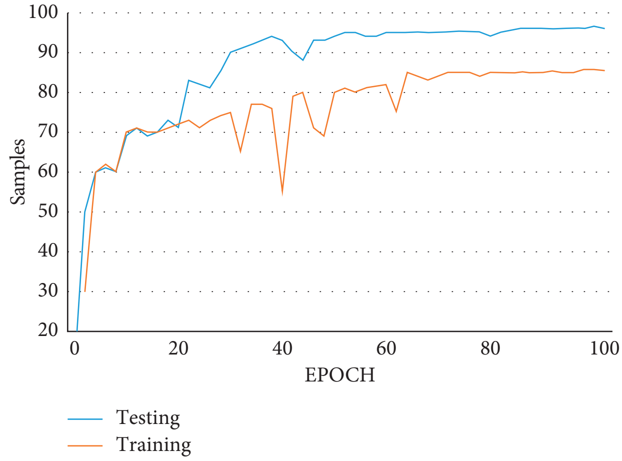


FIGURE 29: Model performance for the training and testing sets.

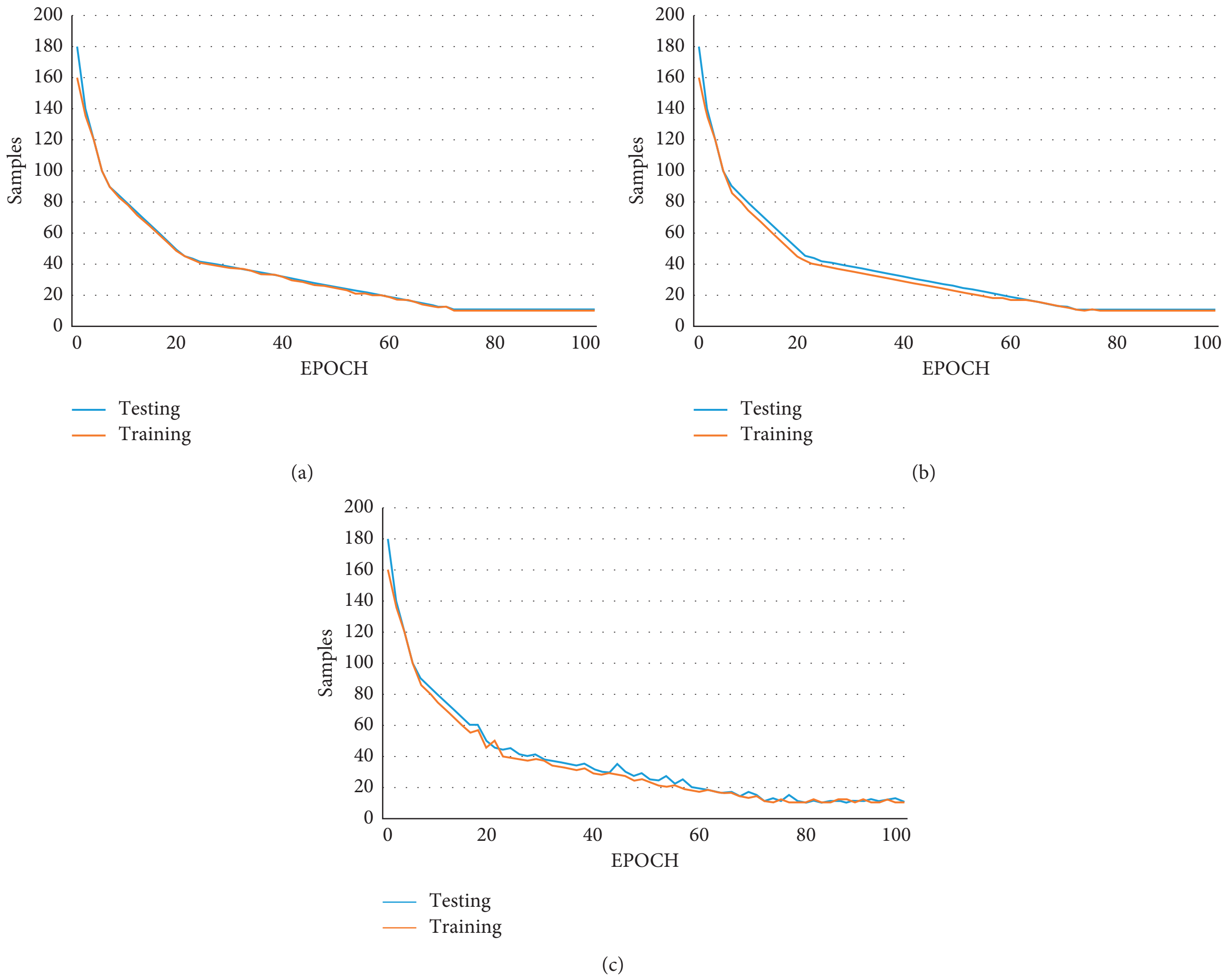


Figure 30: (a) Mean squared error (MSE) loss over training epochs; (b) mean squared logarithmic error (MSLE) loss over training epochs; (c) mean absolute error (MAE) loss over training epochs.

large penalty may affect the performance of the model. To avoid performance issues, the logarithm value for every predicted value is calculated, and then, the MSE is computed to obtain MSLE. MLSE reduces the penalty enforced on the model if a large spread of values is obtained. The same configuration is followed, and the model is tested for widespread values using MSE and MLSE. From Figure 30(b), it can be observed that the MSE loss is significantly higher for the training and testing sets. This indicates that the model may be showing signs of overfitting as there is a significant drop in the beginning and the model starts to recover gradually. Moreover, convergence between the training and the testing set occurs at a later stage.

For cases with large or small values when compared to the mean value, the model might run into outliers. The mean absolute error loss is considered to be suitable for handling outliers. It is used for calculating the absolute difference between the target and the predicted values. In Figure 30(c), the training and the testing set do not converge, and numerous spikes in values are observed, making it not a good fit in the case of outliers.

Figure 31 shows the overall performance evaluation of all the datasets that have been considered for human action recognition. The respective actions and the corresponding classification accuracy are provided for 41 action categories. For the training and testing, the individual actions such as walking, running, jumping bending, waving, jumping jacks, and skipping display better top-1 accuracy rates as the classification matches the target. However, combined actions such as run-fall, walk-sit, and run-jump-walk also show a better classification rate when compared to individual instances. The classification accuracy for standalone actions such as catching, entering, exiting, diving side, horse riding, skate boarding, facial expressions, and wandering was also classified accurately due to the probability of top-5 accuracy as the model considers the top five probabilities that match the target label.

The restricted Boltzmann machine is composed of binary visible units and binary hidden units. The parameters for the RBM are estimated using stochastic maximum likelihood (SML). The time complexity of the RBM network is estimated to be $O(n)$, where n is considered to be the input

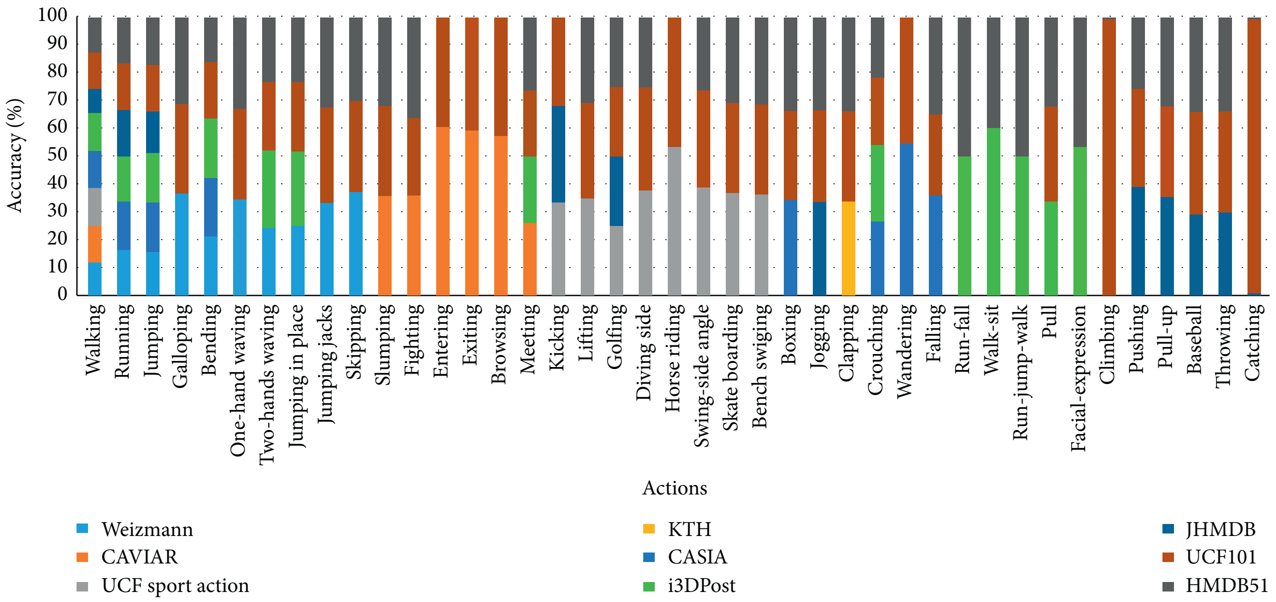


Figure 31: Performance evaluation in terms of accuracy of human action detection.

Table 12: Computational complexity with respect to time for the various datasets.

Dataset	Classes	Resolution	Frames per second (fps)	Input video sample	Training set sample	Testing set sample	Testing accuracy (%)	Training accuracy (%)	Training time	Average epochs
Weizmann	10	180 × 144	25	91	60	31	94.1	96.3	53.05	60
CAVIAR	13	384 × 288	25	29	20	9	97.4	99.1	47.08	40
UCF SA	11	720 × 480	10	150	102	4	94.6	98.2	74.87	39
KTH	6	160 × 120	25	200	140	60	94.52	98.33	102.12	59
CASIA	8	320 × 240	25	250	190	60	95.7	98.75	146.12	75
i3DPost	13	1920 × 1080	15	104	60	44	93.19	97.09	51.91	40
JHMDB	12	320 × 240	25	316	224	92	88.71	90.83	203.72	45
UCF101	101	320 × 240	25	600	400	200	84.21	88.64	283.87	80
HMDB51	51	320 × 240	30	90	60	30	82.11	87.12	54.12	40

features or the number of components. The parameters estimated using SML are the number of components, the learning rate for weight updation, batch size, number of iterations, verbose level, and random state. The random state determines the random number generation for sampling the visible and hidden layers and initializing the components required for sampling the layers during fitting. It also ensures that the data remain uncorrupted, and the scoring sample must obtain accurate results across multiple functions. The attributes considered for training the RBM are the biases of the hidden and visible units; the weight matrix and the hidden activation obtained from the model distribution are computed from the batch size and components.

Table 12 shows the computational complexity with respect to time for the various datasets. The table displays the dataset considered, number of videos, number of classes, pixel resolution, frames per second, the input sample considered for training the model, testing sample, training sample, testing and training accuracy, training time, and average epochs. From Table 12, it can be inferred that the training time increases when the video sample and the pixel resolution increase. The input samples are divided into mini batches and tested with various iterations. The training time after each iteration is recorded, and the time after individual iterations is averaged to obtain the training time of the dataset. The training time for JHMDB and UCF101 datasets is high as the input size and the pixel resolution are high. However, the training times of the datasets can be decreased, and better computation complexity can be achieved with better computational resources.

6. Conclusion

In this paper, a parameter adaptive initialization method that uses a neural network is proposed. The parameter initialization method is modeled based on Maxout activation function using RBM-NN. The spatial and temporal features are learned from various human action datasets. From the experimental analysis, the model learns the spatial-temporal features from the shape feature sequences. An RBM-based neural network model is designed with two layers, and an SVM classifier recognizes multiclass human actions. The proposed method is tested on various benchmark datasets and compared with existing state-of-the-art techniques. The experimental results showed that the proposed method accurately identifies various human actions. The recognition rate was found to be significantly better than other state-of-the-art specific and multiclass human action recognition techniques.

List of Abbreviations

RBM:	Restricted Boltzmann machines
RBM-NN:	Restricted Boltzmann machines-neural network
MAF:	Maxout activation function
SDG:	Stochastic gradient descent
MSE:	Mean squared error
MSLE:	Mean squared logarithmic error
MAE:	Mean absolute error.

Data Availability

The image datasets used to support the findings of this study are included in the article.

Disclosure

The research neither received any funding nor was performed as part of the employment. The research was solely carried by the authors.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

All authors were involved in writing, editing, and proofreading the manuscript.

References

[1] A. F. Bobick and J. W. Davis, "The recognition of human movement using temporal templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 3, pp. 257–267, 2001.

[2] S. Singh, S. A Velastin, and H. Ragheb, "Muhavi: a multicamera human action video dataset for the evaluation of action recognition methods," in *Proceedings of the Advanced Video and Signal Based Surveillance (AVSS)*, pp. 48–55, Boston, MA, USA, September 2010.

[3] M. Ramezani and F. Yaghmaee, "A review on human action analysis in videos for retrieval applications," *Artificial Intelligence Review*, vol. 46, no. 4, pp. 485–514, 2016.

[4] X. Zhai, Y. Peng, and J. Xiao, "Cross-media retrieval by intramedia and inter-media correlation mining," *Multimedia Systems*, vol. 19, no. 5, pp. 395–406, 2013.

[5] K. Tang, L. Fei-Fei, and D. Koller, "Learning latent temporal structure for complex event detection," in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition*, Providence, RI, USA, June 2012.

[6] V. Duong, H. H Bui, D. Q Phung, and S. Venkatesh, "Activity recognition and abnormality detection with the switching hidden semi- markov model," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, San Diego, CA, USA, June 2005.

[7] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, San Diego, CA, USA, 2016.

[8] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two- stream network fusion for video action recognition," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, June 2016.

[9] Y. Kong, S. Gao, B. Sun, and Y. Fu, "Action prediction from videos via memorizing hard-to-predict samples," in *Proceedings of the AAAI*, New Orleans, LA, USA, February 2018.

[10] M. Raptis and L. Sigal, "Poselet key-framing: a model for human activity recognition," in *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*, Portland, OR, USA, June 2013.

[11] I. Laptev, "On space-time interest points," *International journal of computer vision*, vol. 64, no. 2-3, pp. 107–123, 2005.

[12] B. Li Fei-Fei and Li Fei-Fei, "Recognizing human-object interactions in still images by modeling the mutual context of objects and human poses," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 9, pp. 1691–1703, 2012.

[13] J. Donahue, L. Hendricks, S. Guadarrama et al., "Long-term recurrent convolutional networks for visual recognition and description," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, June 2015.

[14] J. Y. Ng, M. Hausknecht, S. Vijayanarasimhan et al., "Beyond short snippets: deep networks for video classification," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, June 2015.

[15] A. Karpathy, G. Toderici, S. Shetty et al., "Large-scale video classification with convolutional neural networks," in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, USA, June 2014.

[16] Y. Yang and M. Shah, "Complex events detection using data-driven concepts," in *Proceedings of the ECCV*, Florence, Italy, December 2012.

[17] Q. Yuan, Q. Zhang, J. Li, H. Shen, and L. Zhang, "Hyperspectral image denoising employing a spatial–spectral deep residual convolutional neural network," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 2, pp. 1205–1218, 2019.

[18] T. Pl otz, N. Y. Hammerla, and P. Olivier, "Feature learning for activity recognition in ubiquitous computing," in *Proceedings of the IJCAI*, Barcelona, CA, USA, July 2011.

[19] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," in *Proceedings of the ICML*, Haifa, Israel, June 2010.

[20] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: a review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[21] Y. Wang, W. Zhou, Q. Zhang, X. Zhu, and H. Li, "Low-Latency human action recognition with weighted multi-region convolutional neural network," in *Proceedings of the CoRR*, Kraków, Poland, September 2018.

[22] G. W. Taylor, R. Fergus, Y. LeCun, and C. Bregler, "Convolutional learning of spatio-temporal features," in *Proceedings of the ECCV*, Crete, Greece, September 2010.

[23] I. Laptev and T. Lindeberg, "Space-time interest points," in *Proceedings of the ICCV*, pp. 432–439, Nice, France, 2003.

[24] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proceedings of the Alvey Vision Conference*, Manchester, UK, September 1988.

[25] J. Carreira and A. Zisserman, "Quo vadis, action recognition? A new model and the kinetics dataset," in *Proceedings of the CVPR*, Honolulu, HI, USA, July 2017.

[26] J. Li, X. Liu, M. Zhang, and D. Wang, "Spatio-temporal deformable 3D ConvNets with attention for action recognition," *Pattern Recognition*, vol. 98, 2020.

[27] F. C. Heilbron, V. Escorcia, B. Ghanem, and J. C. Niebles, "Activitynet: a large-scale video benchmark for human activity understanding," in *Proceedings of the CVPR*, Boston, MA, USA, June 2015.

[28] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile, December 2015.

[29] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 221–231, 2013.

[30] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," in *Proceeding of ICCV*, Beijing, China, October 2005.

[31] E. Mahmoudi and A. Sepahdar, "Exponentiated Weibull-Poisson distribution: model, properties and applications," *Mathematics and Computers in Simulation*, vol. 92, 2012.

[32] C. Yuan, X. Li, W. Hu, H. Ling, and S J Maybank, "3D R transform on spatio-temporal interest points for action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 724–730, Portland, OR, USA, June 2013.

[33] C. Yuan, X. Li, W. Hu, H. Ling, and S. J. Maybank, "Modeling geometric-temporal context with directional pyramid Co-occurrence for action recognition," *IEEE Transactions on Image Processing*, vol. 23, no. 2, pp. 658–672, 2014.

[34] X. Wu, D. Xu, L. Duan, and J. Luo, "Action recognition using context and appearance distribution features," in *Proceedings of the CVPR*, Providence, RI, USA, June 2011.

[35] D. Kim and I. Essa, "Gaussian process regression flow for analysis of motion trajectories," in *Proceedings of the ICCV*, Barcelona, Spain, November 2011.

[36] C. Schu¨ldt, I. Laptev, and B. Caputo, "Recognizing human actions: a local svm approach," in *Proceedings of the IEEE ICPR*, Cambridge, UK, August 2004.

[37] M. Bregonzio, S. Gong, and T. Xiang, "Recognizing action as clouds of space-time interest points," in *Proceedings of the CVPR*, Miami, FL, USA, June 2009.

[38] W. Liu, Y. Li, X. Lin, D. Tao, and Y. Wang, "Hessian-Regularized Co-training for social activity recognition," *PLoS One*, vol. 9, Article ID 108474, 2014.

[39] P. Scovanner, S. Ali, and M. Shah, "A 3-dimensional sift descriptor and its application to action recognition," in *Proceedings of ACM Multimedia*, Brisbane, Australia, October 2007.

[40] A. Klaser, M. Marszalek, and C. Schmid, "A spatio-temporal descriptor based on 3d-gradients," in *Proceedings of the BMVC*, Leeds, UK, September 2008.

[41] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *Proceedings of the CVPR*, Anchorage, AK, USA, June 2008.

[42] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Dense trajectories and motion boundary descriptors for action recognition," *International Journal of Computer Vision*, vol. 103, no. 1, pp. 60–79, 2013.

[43] H. Wang, A. Klaser, C. Schmid, and L. Liu, "Action recognition by dense trajectories," in *Proceedings of the IEEE Conference on Computer Vision & Pattern Recognition*, pp. 3169–3176, Colorado Springs, CO, USA, June 2011.

[44] H. Wang, M. M. Ullah, A. Klaser, I. Laptev, and C. Schmid, "Evaluation of local spatio-temporal features for action recognition," in *Proceedings of the BMVC*, Leeds, UK, September 2008.

[45] M. Liu, H. Liu, Q. Sun, T. Zhang, and R. Ding, "Salient pairwise spatio-temporal interest points for real-time activity recognition," *CAAI Transactions on Intelligent Technology*, vol. 1, no. 1, pp. 14–29, 2016.

[46] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of Imaging Under- Standing Workshop*, Los Angeles, CA, USA, February 1981.

[47] J. Sun, X. Wu, S. Yan, L. Cheong, T. Chua, and J. Li, "Hierarchical spatio-temporal context modeling for action recognition," in *Proceedings of the CVPR*, Miami, FL, USA, June 2009.

[48] M. Jain, H. Je´gou, and P. Bouthemy, "Better exploiting motion for better action recognition," in *Proceedings of the CVPR*, Portland, OR, USA, June 2013.

[49] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Proceedings of the NIPS*, Montreal, Canada, Decemebr 2014.

[50] A. Kar, N. Rai, K. Sikka, and G. Sharma, "Adascan: adaptive scan pooling in deep convolutional neural networks for human action recognition in videos," in *Proceedings of the CVPR*, Honolulu, HI, USA, July 2017.

[51] R. Girdhar, D. Ramanan, A. Gupta, J. Sivic, and B. Russell, "Action-vlad: learning spatio-temporal aggregation for action classification," in *Proceedings of the CVPR*, Honolulu, HI, USA, July 2017.

[52] I. Laptev and P. Perez, "Retrieving actions in movies," in *Proceedings of the ICCV*, Rio de Janeiro, Brazil, June 2007.

[53] D. Tran and A. Sorokin, "Human activity recognition with metric learning," in *Proceedings of the ECCV*, Marseille, France, October 2008.

[54] S. Sobczak, R. Kapela, K. McGuinness et al., "Restricted Boltzmann machine as an aggregation technique for binary descriptors," *Visual Computing*, 2019.

[55] N. Ikizler and D. Forsyth, "Searching video for complex activities with finite state models," in *Proceedings of the CVPR*, Minneapolis, MN, USA, June 2007.

[56] S. Rajko, G. Qian, T. Ingalls, and J. James, "Real-time gesture recognition with minimal training requirements and on-line learning," in *Proceedings of the CVPR*, Minneapolis, MN, USA, June 2007.

[57] S. B. Wang, A. Quattoni, L.-P. Morency, D. Demirdjian, and T. Darrell, "Hidden conditional random fields for gesture recognition," in *Proceedings of the CVPR*, New York, NY, USA, June 2006.

[58] L.-P. Morency, A. Quattoni, and T. Darrell, "Latent-dynamic discriminative models for continuous gesture recognition," in *Proceedings of the CVPR*, Minneapolis, MN, USA, June 2007.

[59] L. Wang and D. Suter, “Recognizing human activities from silhouettes: motion subspace and factorial discriminative graphical model,” in *Proceedings of the CVPR*, Minneapolis, MN, USA, June 2007.

[60] C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas, “Conditional models for contextual human motion recognition,” in *Proceedings of the International Conference on Computer Vision*, Beijing, China, October 2005.

[61] J. C. Niebles, C.-W. Chen, and L. Fei-Fei, “Modeling temporal structure of decomposable motion segments for activity classification,” in *Proceedings of the ECCV*, Crete, Greece, September 2010.

[62] Z. Wang, J. Wang, J. Xiao, K.-H. Lin, and T. S. Huang, “Substructural and boundary modeling for continuous action recognition,” in *Proceedings of the CVPR*, Providence, RI, USA, June 2012.

[63] Q. Shi, L. Cheng, L. Wang, and A. Smola, “Human action segmentation and recognition using discriminative semi-markov models,” *International Journal of Computer Vision*, vol. 93, no. 1, pp. 22–32, 2011.

[64] K. Tang and L. Fei-Fei, “Learning latent temporal structure for complex event detection,” in *Proceedings of the CVPR*, Providence, RI, USA, November 2013.

[65] C. Fanti, L. Zelnik-Manor, and P. Perona, “Hybrid models for human motion recognition,” in *Proceedings of the CVPR*, San Diego, CA, USA, June 2005.

[66] J. C. Niebles and L. Fei-Fei, “A hierarchical model of shape and appearance for human action classification,” in *Proceedings of the CVPR*, Minneapolis, MN, USA, June 2007.

[67] W. Choi, K. Shahid, and S. Savarese, “Learning context for collective activity recognition,” in *Proceedings of the CVPR*, Providence, RI, USA, June 2011.

[68] Y. Kong, Y. Jia, and Y. Fu, “Interactive phrases: semantic descriptions for human interaction recognition,” in *Proceedings of the PAMI*, Columbus, OH, USA, 2014.

[69] G. Luo, S. Yang, G. Tian, C. Yuan, W. Hu, and S. J. Maybank, “Learning human actions by combining global dynamics and local appearance,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 12, pp. 2466–2482, 2014.

[70] C. Yuan, W. Hu, G. Tian, S. Yang, and H. Wang, “Multi-task sparse learning with beta process prior for action recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 423–429, Washington, DC, USA, 2013.

[71] S. Yang, C. Yuan, B. Wu, W. Hu, and F. Wang, “Multi-feature max- margin hierarchical bayesian model for action recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1610–1618, Boston, MA, USA, June 2015.

[72] C. Yuan, B. Wu, X. Li, W. Hu, S. Maybank, and F. Wang, “Fusing $${\mathcal {R}}$$ R features and local features with context-aware kernels for action recognition,” *International Journal of Computer Vision*, vol. 118, no. 2, pp. 151–171, 2016.

[73] D. Wu and L. Shao, “Leveraging hierarchical parametric networks for skeletal joints based action segmentation and recognition,” in *Proceedings of the IEEE CVPR*, pp. 724–731, Columbus, OH, USA, June 2014.

[74] F. Cruciani, A. Vafeiadis, C. Nugent et al., “Feature learning for human activity recognition using convolutional neural networks,” *CCF Transactions on Pervasive Computing and Interaction*, vol. 2, no. 1, pp. 18–32, 2020.

[75] J. Donahue, L. Hendricks, S. Guadarrama et al., “Long-term recurrent convolutional networks for visual recognition and description,” in *Proceedings of the CVPR*, Boston, MA, USA, June 2015.

[76] B. B. Amor, J. Su, and A. Srivastava, “Action recognition using rate-invariant analysis of skeletal shape trajectories,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 1, pp. 1–13, 2016.

[77] T. Qi, Y. Xu, Y. Quan, Y. Wang, and H. Ling, “Image-based action recognition using hint-enhanced deep neural networks,” *Neurocomputing*, vol. 267, pp. 475–488, 2017.

[78] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, “Learning and transferring mid-level image representations using convolutional neural networks,” in *Proceedings of the IEEE CVPR*, pp. 28–37, Columbus, OH, USA, June 2014.

[79] H. Robbins and S. Monro, “A stochastic approximation method,” *The Annals of Mathematical Statistics*, vol. 22, no. 3, pp. 400–407, 1951.

[80] L. Bottou and O. Bousquet, “The tradeoffs of large scale learning,” in *Proceedings in Advanced Neural Information Process and Systems*, pp. 161–168, Vancouver, BC, Canada, December 2008.

[81] Y. Jia, E. Shelhamer, and J. Donahue, “Caffe: convolutional architecture for fast feature embedding,” in *Proceedings of the 22nd ACM International Conference on Multimedia*, pp. 675–678, Mountain View, CA, USA, June 2014.

[82] K. He, X. Zhang, and S. Ren, “Delving deep into rectifiers: surpassing human-level performance on imagenet classification,” in *Proceedings of IEEE international Conference on Computer Vision*, pp. 1026–1034, Santiago, Chile, December 2015.

[83] G. E. Hinton, “A practical guide to training restricted Boltzmann machines,” *Lecture Notes in Computer Science*, Springer, Berlin, Germany, pp. 599–619, 2012.

[84] D. D. Dawn and S. H. Shaikh, “A comprehensive survey of human action recognition with spatio-temporal interest point (STIP) detector,” *The Visual Computer*, vol. 32, no. 3, pp. 289–306, 2016.

[85] Z. Liang, X. Wang, and R. Huang, “An expressive deep model for human action parsing from a single image,” in *Proceedings of International Conference on Multimedia and Expo*, pp. 1–6, Chengdu, China, July 2014.

[86] C. C. Chang and C. Lin, “LIBSVM: a library for support vector machines,” *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 27–33, 2011.

[87] M. D. Rodriguez, J. Ahmed, and M. Shah, “Action mach a spatio-temporal maximum average correlation height filter for action recognition,” in *Proceedings of the Computer Vision and Pattern Recognition*, pp. 15–23, Anchorage, AK, USA, June 2008.

[88] S. Ji, W. Xu, M. Yang, and K. Yu, “3D convolutional neural networks for human action recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 221–231, 2013.

[89] G. E. Hinton and R. Salakhutdinov, “A better way to pretrain deep Boltzmann machines,” in *Proceedings of Advances in Neural Information Processing Systems*, pp. 2447–2455, Lake Tahoe, Nevada, December 2012.

[90] J. Walker, A. Gupta, and M. Hebert, “Patch to the future: unsupervised visual prediction,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3302–3309, Columbus, OH, USA, June 2014.

[91] B. Ziebart, N. Ratliff, G. Gallagher et al., “Planning-based prediction for pedestrians,” in *Proceedings of the IROS*, St. Louis, MO, USA, October 2009.

[92] M. A. Carreira-Perpinan and G. E. Hinton, "On contrastive divergence learning," in *Proceedings of the CCVP*, pp. 97–106, Grote Baan, Belgium, 2005.

[93] Y. Kong, Z. Tao, and Y. Fu, "Deep sequential context networks for action prediction," in *Proceedings of the CVPR*, Honolulu, HI, USA, July 2017.

[94] I. J. Goodfellow, D. Warde-Farley, and M. Mirza, "Maxout networks," 2013, http://arxiv.org/abs/1302.4389.

[95] T.-Y. Lin, "Microsoft COCO: Common objects in context," *Lecture Notes in Computer Science*, p. 8693, Springer, Berlin, Germany, 2014.

[96] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: a large-scale hierarchical image database," in *Proceedings of the CVPR*, Miami, FL, USA, June 2009.

[97] A. Krizhevsky, "Learning multiple layers of features from tiny images," Technical Report TR-2009, University of Toronto, Toronto, Canada, 2009.

[98] C. Xu, J. Yang, and J. Gao, "Coupled-learning convolutional neural networks for object recognition," *Multimedia Tools and Applications*, vol. 78, no. 1, pp. 573–589, 2019.

[99] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 12, pp. 2247–2253, 2007.

[100] D. S. Cheng, M. Cristani, M. Stoppa, L. Bazzani, and V. Murino, "Custom pictorial structures for re-identification," in *Proceedings of the BMVC*, Dundee, UK, August 2011.

[101] K. Soomro and A. R. Zamir, "Action recognition in realistic sports videos," *Computer Vision in Sports*, Springer International Publishing, New York, NY, USA, 2014.

[102] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: a local SVM approach," in *Proceedings of ICPR'04*, Cambridge, UK, August 2004.

[103] Center for Biometrics and Security Research, *CASIA Action Database for Recognition*, Center for Biometrics and Security Research, 2011.

[104] N. Gkalelis, H. Kim, A. Hilton, N. Nikolaidis, and I. Pitas, "The i3DPost multi-view and 3D human action/interaction," in *Proceedings of the CVMP*, pp. 159–168, London, UK, November 2009.

[105] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M. J. Black, "Towards understanding action recognition," in *Proceedings of the International Conference on Computer Vision (ICCV)*, pp. 3192–3199, Sydney, Australia, December 2013.

[106] K. Soomro, A. Roshan Zamir, and M. Shah, "UCF101: a dataset of 101 human action classes from videos in the wild," 2012, http://arxiv.org/abs/CRCV-TR-12-01.

[107] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB: a large video database for human motion recognition," in *Proceedings of the ICCV*, Barcelona, Spain, November 2011.

[108] M. Raptis, I. Kokkinos, and S. Soatto, "Discovering discriminative action parts from mid-level video representations," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1242–1249, Providence, RI, USA, June 2012.

[109] Y. Kong and Y. Fu, "Max-Margin action prediction machine," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 9, pp. 1844–1858, 2016.

[110] H. A. Abdul-Azim and E. E. Hemayed, "Human action recognition using trajectory-based representation," *Egyptian Informatics Journal*, vol. 16, no. 2, pp. 187–198, 2015.

[111] M. S. Ryoo, "Human activity prediction: early recognition of ongoing activities from streaming videos," in *Proceedings of the ICCV*, Barcelona, Spain, November 2011.

[112] P. Negri and P. Lotito, "Pedestrian detection on CAVIAR dataset using a movement feature space," in *Proceedings of the 13th Argentine Symposium on Technology*, pp. 1850–2806, Tel Aviv, Israel, 2012.

[113] I. Mironică, I. C. Duţă, and B. Ionescu, "A modified vector of locally aggregated descriptors approach for fast video classification," *Multimedia Tools and Applications*, vol. 75, no. 15, pp. 9045–9072, 2016.

[114] Q. V. Le, W. Y. Zou, and S. Y. Yeung, "Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis," in *Proceedings of the IEEE Computer Vision and Pattern Recognition*, pp. 3361–3368, Providence, RI, USA, June 2011.

[115] F. Rezazadegan, S. Shirazi, and N. Sünderhauf, "Enhancing human action recognition with region proposals," *Obstetrics and Gynecology*, vol. 32, no. 3, pp. 335–340, 2015.

[116] N. Souly and M. Shah, "Visual saliency detection using group lasso regularization in videos of natural scenes," *International Journal of Computer Vision*, vol. 117, no. 1, pp. 93–110, 2016.

[117] S. Lai, W.-S. Zheng, J.-F. Hu, and J. Zhang, "Global-local temporal saliency action prediction," *IEEE Transactions on Image Processing*, vol. 27, no. 5, pp. 2272–2285, 2018.

[118] M. Sreeraj, "Multi-posture human detection based on hybrid HOG-BO," in *Proceedings of the Fifth International Conference on Advances in Computing and Communications (ICACC)*, IEEE, Kochi, India, September 2015.

[119] J. Yang, Z. Ma, and M. Xie, "Action recognition based on multi-scale oriented neighborhood features," *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 8, no. 1, pp. 241–254, 2015.

[120] H. Liu, "Study of human action recognition based on improved spatio-temporal features," *Human Motion Sensing and Recognition*, Springer, Berlin, Germany, pp. 233–250, 2015.

[121] L. Shao, "Spatio-temporal Laplacian pyramid coding for action recognition," *IEEE Transactions on Cybernetics*, vol. 44, no. 6, pp. 817–827.

[122] S. Cheng, "Action recognition based on spatio-temporal log-Euclidean covariance matrix," *International Journal of Signal Processing*, vol. 9, no. 2, pp. 95–106, 2014.

[123] M. Sharif, M. A. Khan, and T. Akram, "A framework of human detection and action recognition based on uniform segmentation and combination of Euclidean distance and joint entropy-based features selection," *EURASIP Journal of Image Video Processing*, vol. 2017, p. 89, 2017.

[124] N. Gkalelis, N. Nikolaidis, and I. Pitas, "View indepedent human movement recognition from multi-view video exploiting a circular invariant posture representation," in *Proceedings of the ICME*, New York, NY, USA, June-July 2009.

[125] A. Iosifidis, N. Nikolaidis, and I. Pitas, "Movement recognition exploiting multi-view information," in *Proceedings of the MMSP*, Saint Malo, France, October 2010.

[126] M. Holte, T. Moeslund, N. Nikolaidis, and I. Pitas, "3D human action recognition for multi-view camera systems," in *Proceedings of the 3DIMPVT*, Hangzhou, China, May 2011.

[127] M. Khalid and J. Usman, "Multi-modal three-stream network for action recognition," in *Proceedings of the 24th*

International Conference on Pattern Recognition (ICPR), pp. 3210–3215, Beijing, China, August 2015.

[128] L. Fei-Fei, R. Fergus, and P. Perona, “Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories,” in *Proceedings of the Workshop on Generative-Model Based Vision*, Washington, DC, USA, June-July 2004.

[129] Y. Cao, D. Barrett, A. Barbu et al., “Recognizing human activities from partially observed videos,” in *Proceedings of the CVPR*, Portland, OR, USA, June 2013.

[130] Y. Kong, D. Kit, and Y. Fu, “A discriminative model with multiple temporal scales for action prediction,” in *Proceedings of the ECCV*, Zurich, Switzerland, September 2014.

[131] Y. G. Jiang, Q. Dai, X. Xue, W. Liu, and C. W. Ngo, “Trajectory-based modeling of human actions with motion reference points,” in *Proceedings of the ECCV*, Florence, Italy, October 2012.

[132] H. Wang and C. Schmid, “Action recognition with improved trajectories,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Sydney, NSW, Australia, December 2013.

[133] B. Zhang, L. Wang, and Z. Wang, “Real-time action recognition with enhanced motion vector CNNs,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2718–2726, Las Vegas, NV, USA, June 2016.

[134] H. Wang and C. Schmid, “Action recognition with improved trajectories,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3551–3558, Sydney, NSW, Australia, December 2013.

[135] K. Simonyan and A. Zisserman, “Two-Stream convolutional networks for action recognition in videos,” in *Proceedings of the 27th International Conference on Neural Information Processing Systems*, vol. 1, pp. 568–576, Montreal, Canada, December 2014.