# SPATIO-TEMPORAL DEPTH MOTION DESCRIPTOR FOR ACTION RECOGNITION UISNG 2D CONVOLUTIONAL NEURAL NETWORKS

**S. Sandhya Rani[1], Dr. G. Appa Rao Naidu[2],Dr.V. Usha Shree[3]**

[1]Research Scholar, Department of CSE,JNTUH, India

[2]Professor, Department of CSE,JBIET, India

[3]Principal, Department of ECE, JBREC, India

**Abstract:** *This paper proposes a new Human Action Recognition Framework by fusing the Motion History with Depth Motion information. Firstly, Difference Depth Motion Map ($D^2MM$) is proposed to capture the shape and motion cues of action. Particularly, $D^2MM$ nullifies the side effects like body shaking movements and Ghost Shadows. Secondly, Modified Motion History Image ($M^2HI$) is proposed to identify the real ad fake movements in human body and to encode motion information effectively. Before the action representation, this work employed temporal segmentation in which the entire sequence of frames of a depth video is segmented into different sets with different frame lengths. Then the proposed two action representation methods are applied over the segments. For feature extraction and classification, we have employed a simple 2D Convolutional Neural Network (2D-CNN) in two phases and the results are fused to obtain final classification score. Simulation experiments are conducted over MSR action 3D dataset and MSR daily activity 3D dataset, shows the effectiveness of proposed method.*

Keywords: Action Recognition, Depth data, Depth Motion Map, Convolutional Neural Network, Late fusion, MSR action dataset, Accuracy.

## I. INTRODUCTION

In recent years, vision based action recognition has gained a great research interest due to its widespread applicability in several applications including context based video retrieval, autonomous driving vehicle, human-computer interactions, intelligent surveillance systems [1] [2], and gaming applications [3]. In earlier several approaches are proposed to recognize human action but most of them considered the traditional RGB videos which were captured through normal cameras [4], [5]. However, the traditional RGB videos have several constraints such as different lighting conditions, varying backgrounds, color and textural variations etc., due to which the action recognition has become a challenging task in computer vision.

In recent years, with the advent of low-cost depth cameras like Microsoft Kinect [6], the action recognition research has entered into a new phase in which the input videos of an action recognition system are RGB Depth (RGB-D) videos. The depth cameras ensure depth data as well as color images sequence in real time, which makes the action recognition system more realistic and solves the traditional problems with RGB videos. The RGB-D videos are more advantageous than the traditional RGB videos in several aspects: (1) the depth cues in RGB-D videos are insensitive to illuminations variations and they can capture the videos even in dark environments also. (2) Depth videos can provide depth data while the traditional ones can't. (3) Texture and color variations are not present in the depth videos which make the action unit detection easier [7].

Considering the advantages of depth videos, an extensive research effort has been made to achieve better recognition results. Among the earlier developed action representation methods, Depth Motion Maps (DMM) [8] and Motion History Images (MHI) [9] have gained better performance in the recognition of human actions. However, they are susceptible for cluttered backgrounds, small body shaking movements, jumbled objects and low resolution depth videos. Moreover, in the depth map construction, the DMM process considers the entire frames of depth video by which the detailed temporal information in the subset of images can't be captured.

To solve these problems, in this paper, we have developed a new action recognition framework based on the combination of DMM and MHI. Two new action representation schemes namely, Difference Depth Motion Map ($D^2MM$) and Modified Motion History Image ($M^2HI$) are proposed to represent the 3D depth action video in 2D image format. Before action representation, to acquire detailed motion information, this paper employed temporal segmentation at different levels. Further for feature extraction and classification we have employed 2D Convolutional Neural Network (2D-CNN) model and the final action label is evaluated based on late fusion. At the late fusion, we have considered three different fusion techniques such as Maximum fusion, Product fusion and linear fusion. Simulation experiments are conducted over MSR action 3D dataset and MSR daily activity 3D dataset and the performance is measured through recognition accuracy.

Remaining paper is organized as follows; Section II demonstrates the details of literature survey. Section III explores the details of proposed action recognition model. Simulation experiments details are demonstrated in section IV and the concluding remarks are shown in section V.

## II. LITERATURE SURVEY

Due to the great success of depth action video sin action recognition, lot of research works has been conducted in this orientation. A detailed literature survey on action recognition methods can be found at [10-12]. In this section, the recent works related to the paper, including, action segmentation, and action representation are briefly reviewed.

### A. Segmentation

Generally, most of the earlier developed action recognition works have assumed that the individual instances of action in the frame or image are already isolated or segmented. In the case of continuous action recognition from depth videos, the input stream generally contains unknown number of frames, unknown number of objects, and unknown boundaries of actions and to this problem has to be solved at the same time. Means, the feature extraction or action representation or action recognition method has to solve this problem, which is highly complex and also results in less quality.

With respect to action segmentation, Dynamic Time Wrapping (DTW) is one of the widely employed techniques to determine the delimiting frames of individual actions [13], [14]. In this approach, for a given action sequence, initially difference images are extracted by subtracting the successive grayscale images. Next, the difference images are divided into small blocks of size $3 \times 3$ and every block is represented with an average value of pixels within the block. Then the final feature of each frame, called as motion feature is calculated by flattening the blocks in the difference image, excluding the final frame. This process is accompanied for both training and testing videos. Finally the action is segmented by comparing the motion features and analyzed through Viterbi algorithm [15]. However, the DTW based segmentation makes it difficult to obtain more distinct feature when a filter with small size is applied over the images. Let's consider two different difference images of two different actions. If we apply a filter with size $3 \times 3$ over them, then the obtained motion features may not differ much because at the starting, the two action frames have similar features.

Another possible method of action segmentation is based on appearance. Based on the general assumption about the similar characteristics of starting and ending frames, D. Wu et al., [16] accomplished Histogram of Gradient (HoG) with correlation coefficient and K-Nearest Neighbor (K-NN) [17] to identify the first and last frames of an action video.

Though these methods applied segmentation for action detection, they are not focused over the detection of key frames. All of these methods are applied spatial segmentation which resides within the frame but not reveals about the temporal motion information between frames. Every actor has his/her own speed and ends up the video with varying number of frames. For example, a slow action is completed in larger time and acquired in larger number of frames, which creates a he computational burden over further processes. Hence the frames are need to be segmented temporally through the temporal motion information will also get preserved.

### B. Action representation

Most of the previous works focused on the extraction of Spatio-temporal features to represent an action in depth videos. The most popular approaches are HOG3D [18], 3D EMOSIFT [19], Acionlet [38], HON4D [20], Depth Motion Maps (DMM) [23] and Motion History Images (MHI) [30]. In particular, Yang et al., [8] employed DMMs to acquire cues of capture motion from dissimilar viewpoints (top, side, and front) and they employed HOGs to represent an action. Similarly, Chen et al., [21] accomplished Local Binary patterns for feature extraction after representing the depth action video with DMM from three projection views such as Front view, side view and top view. Kernel Based Extreme Learning Machine (KELM) is employed for action

classification. This method also had shown its contribution at fusion level by applying two levels of fusions such as feature level fusion and decision level fusion. At feature level the LBP features are fused and at decision level, soft max rule is applied to combine classification scores. Further, the same author, Chen et al., [22] generated the DMMs are segment level at which the depth video sequence is segmented into several overlapping segments. Then each segment is characterized by DMM followed by LBP to extract the location rotation invariant information. In the final stage, Fisher kernel is accomplished to generate a compact feature vector for every action. ELM is applied for action classification.

Next, focusing over the segmentation and motion information, M. A1-faris et al., [24] proposed a new version of DMM, called as "Fuzzy weighted multi-resolution DMMs (FWMDMMs)". This model focused over the creation of multiple DMMS at multiple levels by segmenting the temporal action frames at different levels. After DMMs representation, to find the significance, this method employed fuzzy weight function in three orientations such as Linear, reverse and central. Finally the FWMDMMs are fed to deep CNN model for classification [25]. Further, the main aim of Xu Weiyao et al., [26] is to exploit the frame selection and towards such, a new "Multilevel Frame Select Sampling (MFSS)" method is proposed to generate three levels of temporal samples from input depth sequences. Then they are represented through Motion and Static Mapping (MSM) followed by Block based and LBP and Fisher kernel representation. KELM is accomplished for action classification. Further extending the LBP to "Discriminative Completed LBP (DiscLBP)", Wu Li et a., [27] proposed DMM assisted action recognition with two classifiers such as ELM and collaborative representation classification (CRC).

Next, MHI is also a basic representation of an action image which can also determines the motion information at every pixel. Based on this fact, Watanabe et al., [28] extracted shift invariant features based on "Higher Order Local Correlation (HOLC)" from the MHI for action recognition. Next, Y.L. Tian et al., [29] considered MHIs as a basic action representation and over the obtained MHI they employed a global and location filtration method to eliminate the unaltered motions. "Gaussian Mixture Model (GMM)" based classifier is employed for action classification. Next, combining the Static History Image with MHI, E. Chen et al., [31] proposed an action recognition framework in which LBP is applied for feature extraction. Support Vector Machine is accomplished for classification. Further, D. Kim et al., [32] projected the depth map into two orthogonal planes such as front and side view. And then applied two descriptors namely, "Depth Motion History (DMH)" and "Depth Motion Appearance (DMA)" for action representation and finally applied SVM for classification.

## III. PROPSOED APPROACH

### 3.1 Overview

In this paper, we have a novel deep learning based action recognition framework to recognize human actions form depth action videos. This framework employs Spatio-temporal features from depth videos for action recognition. Here the Spatio-temporal data is incorporated with two different action representation techniques: Temporal Difference Depth Motion Map ($TD^2MM$) and Temporal Modified Motion History Image ($TM^2HI$). Each of the method provides explicit action movements in the form of certain shape and motion cues. Further the feature extraction and classification is employed through 2D-Convolutional Neural Network (2D-CNN). Two 2D-CNN models are trained on these two action representations and the obtained classification results are fused to derive the ultimate classification label. Figure.1 shows the block diagram of proposed action recognition framework. The major contributions of this work are outlined as follows;

1. We didn't rely on the handcrafted features through which the system will consume additional time to extract the features form every action model. Instead of that we directly used CNN to extract the Spatio-temporal information from data.

2. Unlike the conventional DMM which consider all the frames, we have considered only few set of frames for DMMs construction. Moreover, the proposed $TD^2MM$ nullifies the extra side effects like ghost shadows, and small body shake movements.

3. Unlike the conventional MHI which can't determine the whether the pixel belongs to the moving portion of human body or a fake moving pixel, the proposed $TM^2HI$ can differentiate between these two effectively.

4. At decision level, we have employed late fusion get the final decision regarding classification label. This makes the recognition system to train different models simultaneously.

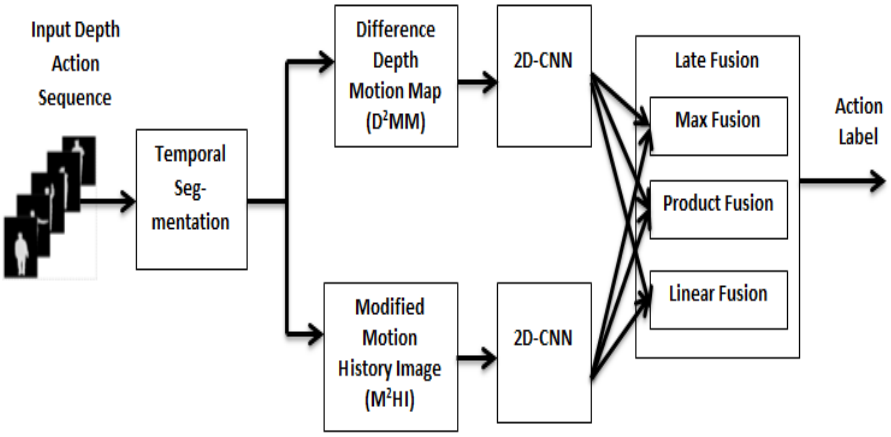



**Figure.1 Block diagram of proposed action recognition model**

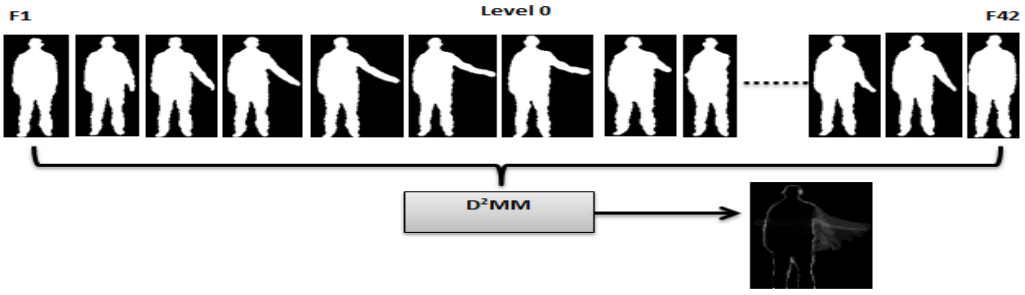
### 3.2 Action Representation

Action representation is an important task in the action recognition model. Representing an action with motion information is one of the significant way in which an entire video can be represented in a single 2D image. In our model, we have employed two different action representation techniques such as $TD^2MM$ and $TM^2HI$. DMM and MHI are the base sources of these two techniques which have gained significant importance in the action representation with motion information. DMM and MHI are sensitive to noise caused by several side effects such as cluttered background, jumbled objects and low resolution cameras. Hence the DMM is extended to $D^2MM$ and MHI is extended to $M^2HI$ to remove the external noises thus the quality of DMM and MHI will get enhanced.

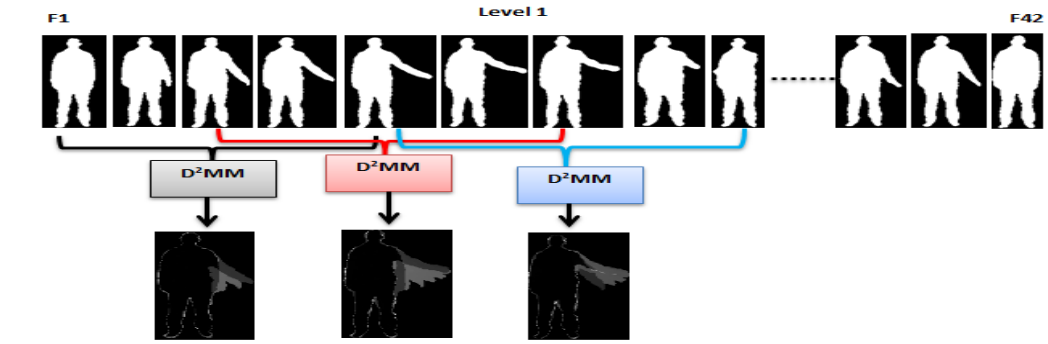
#### 3.2.1 $TD^2MM$

DMM has gained much importance in the representation of human actions in depth videos. In DMM, the 3D action video is transformed into a 2D motion image and has been successfully applied for action recognition. DMMs are basically used to determine the motion and shape information of a depth action video. Generally the DMM is obtained by the accumulation of difference between the frames of an action sequence. Actually the DMM is constructed by considering all the frames into the process. However, this process may not be able to capture more and detailed temporal information. Hence to capture more and effective motion information, we have employed temporal segmentation in which the depth video sequence is divided into several temporal and overlapped set of segments with equivalent number of frames in every set. Then DMM is applied over these segments to obtain 2D motion image. Since the actions performed by different people have different variations like speed and number, consideration of entire frames or equal number of frames for computation of DMM is not a suggestible technique. Hence we have employed three different levels and at level, we have considered different number of frames for DMM computation. Further, we have considered a uniform overlapped frames to segment the video sequence into several sets of segments, i.e., the segmentation will maintain a constant number of overlapping frames between every two successive segments.

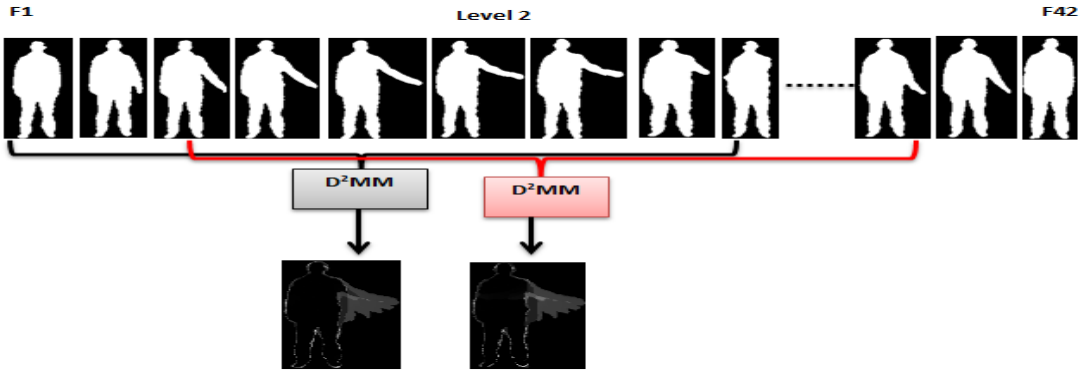
Under this temporal segmentation, we have considered tally three levels such as $L_0, L_1$ and $L_2$. At the level $L_0$, the process considers entire set of frames for depth map constriction. Next at level $L_1$, the process considers five frames with an overlapping factor R = 3. Further at level $L_2$, the process considers ten frames with same overlapping factor. A simple demonstration about temporal segmentation is depicted in figure.2.




(a)




(b)




(c)

**Figure.2 Temporal Segmentation (a) Level 0, (b) Level 1 and (c) Level 3**

As shown in figure.2, the depth video sequence is composed of 42 frames. Figure.2(a) demonstrates the original depth map construction at level $L_0$ in which the entire set of frame are considered. Next, Figure.2(b) shows the depth map construction at level $L_1$ in which the entire action sequence is segmented in several sets and every set is composed of five frames. Here we have employed the overlapping factor as 3. For example, the first set consists of frames starting from F1 to F5, the second set consists of frames starting from F3 to F7. Hence the common frames between set 1 and set 2 are F3, F4 and F5. In this way entire sequence is segmented at level $L_1$. Next, at level $L_2$, the entire action sequence is segmented in several sets and every set is composed of five frames, as shown in Figure.2(c). Here also we have employed the overlapping factor as 3. For example, the first set consists of frames starting from F1 to F10, the second set consists of frames starting from F3 to F12. Hence the common frames between set 1 and set 2 starts from F3, and ends at F10. In this way entire sequence is segmented at level $L_2$.

Once the frames are segmented into different segments, then they are subjected to depth map construction through $D^2$MM according to the method described in [33]. Even though the traditional DMM can acquire shape and motion cues of an action effectively, several side effects present in the depth action sequence like ghost shadows, and small body shaking movements' results in some unidentifiable energy regions in DMM. To suppress these external regions, $D^2$MM initially computes a binary motion image based the difference between consecutive frames. Next, it measures a weighted motion score of a binary image through sliding window and based on that score, a new difference map is constructed. Finally the $D^2$MM is obtained by the accumulation of all the difference maps. An example output of DMM and $D^2$MM is shown in figure.3.

(a) (b) (c)

**Figure.3 (a) Original Depth action frame, (b) DMM and (c) D$^2$MM**

**3.1.2TM$^2$HI**

Similar to the temporal segmentation employed at TD$^2$MM, here also initially the input depth action sequence is segmented into different segments at various levels. Once the temporal segments are obtained, then they are subjected to M$^2$HI, an extension of tradition MHI. MHI was initially introduced by Bobick and Davis [34]. MHI records the temporal variations history at every pixel in the action frames of a video. MHI is a very simple and effective encoding scheme that encodes the spatial distributions of movements of an action. For this computation, MHI considers the pixel intensities of a pixel located at *(x, y)* in a temporal frame at time *t, (x, y, t).* The proposed M$^2$HI is employed over grayscale images because the grayscale images preserve the compact motion information and also makes the M$^2$HI less sensitive to noise and illumination variations. Consider an RGB-D action image sequence, with N frames, initially it is converted into grayscale image sequence and let it be $F(x,y,t), t = 0,1, \dots N-1$. Next, convert the grayscale image sequence into binary motion image sequence $B_m(x,y,t), t = 0,1, \dots N-1$, by the computation of difference between consecutive frames $F(x,y,t)$ and $F(x,y,t+1)$. Mathematically the $B_m(x,y,t)$ is obtained as;

$$B_m(x,y,t) = \begin{cases} 1, & if\ |F(x,y,t) - F(x,y,t+1)| > \tau \\ 0, & Otherwise \end{cases} \quad (1)$$

Where $F(x,y,t)$ is the pixel located at position $(x,y)$ in a frame F at time *t*, $F(x,y,t+1)$ is the pixel located at position $(x,y)$ in a frame F at time *t+1*. The Binary motion image can be simply defined as a temporal difference image which has larger deviations in pixel intensities. Next, $\tau$ is a pre-determined threshold that signifies the motion variations. For larger value of $\tau$, the binary motion image obtains only the pixel which have larger deviations in their pixel intensities in the consecutive frames and vice versa. Hence the value of $\tau$ is to be determined experimentally. In our work, we have set the value of $\tau$ as 40, that means the pixels in the binary motion image will become 1 if the difference between the pixels at the same location in two consecutive frames will be greater than 40. Based on the obtained binary motion image, the MHI is computed as;

$$MHI = \sum_{t=0}^{N-2} \omega_t . B_m(x,y,t) \quad (2)$$

Where $\omega_t$ is the weight of a $B_m(x,y,t)$ at time t and is computed as

$$\omega_t = (t+1).{}^{255}/_{N} \quad (3)$$

Here $\omega_t$ follows a linearly increasing characteristic with time *t*. It signifies that the recent binary motion image has higher significance than the oldest binary motion image. This is valid only for some times because as the time progresses, the action will reach to peak and then slowly ends up. After the peak frame, the motion won't have any significance. Moreover, the frames after the peak frame won't contribute much motion information. Hence consideration of an entire set of frames for MHI construction won't have much significance in the provision of more motion information. This problem is solved by considering only a set of frames which is handled through eh temporal segmentation as described in the above subsection.

**Note:** the temporal segmentation is done up to 3 levels only because more number of levels constitutes an increased computational burden and less number of levels can't explore motion information effectively. Further, the speed of an action is also unknown, hence we have employed only up to three levels. For an action with high speed, the entire motion information is covered with less number of frames and vice versa. The segmented sets of frames can achieve better results in both cases due to the multi-level temporal segmentation.

In the traditional MHI, the motion region is identified through pixel-wise difference which results in some unidentified regions due to object boundaries and small body shaking movements. Such kind of undefined regions are treated as noises in the binary image sequence $B_m(x,y,t), t = 0,1, \dots, N-2$, resulting in a poor

quality MHI. To solve this problem, we have proposed a simple and effective method to remove these noises in the $B_m(x,y,t), t = 0,1,\dots,N-2$, called as Modified Motion History image ($M^2HI$). $M^2HI$ employs a 2D sliding window at every pixel location $(x,y,t)$ to decide whether the pixel belongs to the moving portion of human body or a unconnected fake moving pixel. Generally in the action image, the movement of human body occupies a larger number of pixels which results in a more number of connected pixels. In the case of binary image, the real movement of human body occupies a larger region with no-zero pixels. In other words, if the movement at pixel location $(x,y,t)$ is a real movement (having pixel intensity 1), then the neighbor pixels within the 2D sliding window also have pixel intensities 1 only. Otherwise if the pixel at location $(x,y,t)$ is not a real movement and if it is fake movement, then then relative pixels in its neighborhood won't have any significance, i.e., they will be 0's. to compute the motion significance of a pixel at location $(x,y,t)$, we have computed a new score called significant motion score ($S_M(x,y,t)$) for every moving pixel in the binary image $B_m(x,y,t)$, as follows;

$$S_M(x,y,t) = \frac{1}{(w+1)^2}\sum_{i=x-(w/2)}^{x+(w/2)}\sum_{j=y-(w/2)}^{y+(w/2)} B_M(i,j,t) \qquad (4)$$

Where $w$ is the size of sliding window, i.e., height and width. In our work, we have set the height and width value as 6, i.e., $w = 6$. After the computation of significant motion score at every pixel, then the $S_M(x,y,t)$ is checked for the significance, i.e., whether it is significant or not. This is done by comparing the $S_M(x,y,t)$ with a pre-defined threshold $\varphi$. Generally, for a fake moving pixe1, the neighbor pixels differ and the $S_M(x,y,t)$will be close to 0. Unlike, for a real movement, the neighbor pixels also have same value and the $S_M(x,y,t)$ value will be close to 1. But we can't consider only the pixels with $S_M(x,y,t) = 1$ as significant because some real movements are also exists which have $S_M(x,y,t) \approx 1\ (> 0.6)$. Hence, we have employed a threshold based decision to find the some more pixels with real movements which have $S_M(x,y,t) > \varphi$. In this phase, the $S_M(x,y,t)$ is compared with the predefined threshold, $\varphi$. If it is found to be greater than the threshold, then that pixel value is assigned as 1 otherwise zero. Based on this strategy, a new binary motion image $B_M'(x,y,t)$ is computed as follows;

$$B_M'(x,y,t) = \begin{cases} 1, & if\ B_m(x,y,t) = 1\ and\ S_M(x,y,t) > \varphi \\ 0, & otherwise \end{cases} \qquad (5)$$

In our experiments, we have set the threshold $\varphi = 0.6$. Further the new MHI, i.e., $M^2HI$ is constructed based on the obtained new binary motion image $B_M'(x,y,t), t = 0,1,\dots,N-2$ using Eq.(2). The output of this process is a grayscale image with only real movements of the human body where the real moving pixels are more brighter and remaining are less brighter. Figure.4 shows an example of outputs obtained through MHI and $M^2HI$.

### 3.3 CNN Model

Once the human action representation is completed through $TD^2MM$ and $TM^2HI$, the next step is to extract Spatio-temporal features. Here to extract such type of features, we have employed a CNN model according to the CNN model described in our earlier work [33]. The architecture of CNN model is shown in figure.5 which consist of five convolutional layers ad three pooling layers. The structural details such as number of filters and filter size of every layer are described in table.1.

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

**Figure.4 (a) Original Depth action frame, (b) MHI and (c) $M^2HI$**

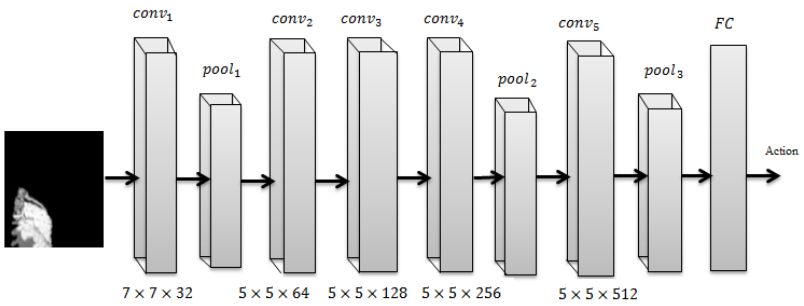



**Figure.5 CNN model for action recognition**

Table.1 CNN structures

| Layer | Filter Size | Stride | Pad |
|---|---|---|---|
| $\boldsymbol{Conv_1}$ | $7 \times 7$ | $2 \times 2$ | 0 |
| $\boldsymbol{Pool_1}$ | - | $2 \times 2$ | - |
| $\boldsymbol{Conv_2}$ | $5 \times 5$ | $1 \times 1$ | 0 |
| $\boldsymbol{Conv_3}$ | $5 \times 5$ | $1 \times 1$ | 0 |
| $\boldsymbol{Conv_4}$ | $5 \times 5$ | $1 \times 1$ | 0 |
| $\boldsymbol{Pool_2}$ | - | $2 \times 2$ | - |
| $\boldsymbol{Conv_5}$ | $3 \times 3$ | $2 \times 2$ | 0 |
| $\boldsymbol{Pool_3}$ | - | $2 \times 2$ | - |

### 3.4 Fusion for Classification

Information fusion is an important aspect in multi-modal action recognition. Basically, the fusion strategies are of three types and they are defined based on the stage at which the fusion is employed. The three types are: Early fusion (fusion of multiple data modals), Intermediate fusion (fusion of multiple features) and late fusion (fusion of multiple decisions). The early fusion needs multi modal data which have some distinct features, which is not our case since we have employed only one model. Next, the intermediate fusion is employed a feature level and it fuses different feature extracted through different feature extraction methods. This is also not suitable for our work because we didn't apply any feature extraction techniques. We have employed different action representations only. The feature extraction is done with CNN only. If we employ intermediate fusion, we have to fuse the motion maps and motion history image obtained through TD$^2$MM and TM$^2$HI, but the size of the feature maps obtained are of very large (almost equal to the size of original frame). This process required a huge memory and also this constitutes a more computation burden at classification level. Hence we have adopted late fusion and here we compute the probabilities for final fusion result calculation. The main advantage of late fusion is the possibility of simultaneous training of data of different feature maps.

In general, late fusion can be employed in two ways: unsupervised fashion and supervised fashion [34]. The supervised decision necessitates an extra training on the obtained outputs at different models. Moreover, according to [35], an additional training of classifier necessitates an extra memory usage and some distinct parameters are needed to set, which is not suitable for our work. SVM and Neural Networks (NN) are the best example for supervised learning. Unlike the supervised learning, the unsupervised learning didn't require any additional training. Majority voting is a best example for unsupervised decision strategy. In unsupervised decision making, the additional memory and additional parameter setting won't come into picture. Hence we have employed an unsupervised decision making strategy to find the final classification score.

Let's consider an action A, each of CNN model produces for each class $C_l, 1 \leq l \leq C$, a class membership probability $P(C_l|A)$. Further let's assume the probability of class membership obtained through TD$^2$MM + CNN is $P_{CD}(C_l|A^{CD})$ and the probability of class membership obtained through TM$^2$HI + CNN is $P_{CM}(C_l|A^{CM})$. Then the final probability of class membership is derived by providing a simple linear relation between $P_{CD}(C_l|A^{CD})$ and $P_{CM}(C_l|A^{CM})$, as

$$P(C_l|A) = \alpha . P_{CD}(C_l|A^{CD}) + (1 - \alpha) . P_{CM}(C_l|A^{CM}) \qquad (6)$$

Here the additional parameter $\alpha$ controls the significance of each model and it is determined experimentally. Along with this fusion, we also have employed two additional and most popular fusions such as product fusion and maximum fusion. Based on the obtained fused results, the final class label $\hat{C}_l$ is determined as the one which have maximum probability of class membership, obtained as;

$$\hat{C}_l = \text{argmax}_{1 \leq l \leq C}\left(P(C_l|A)\right) \quad (7)$$

## IV. SIMULATION EXPERIMENTS

To evaluate the proposed action recognition model, we have used totally two standard datasets and the performance is measured at several instances. Initially this section describes the details of datasets. Next, the details of performance metrics such as Recall, Positive Predictive Value (PPV), F-Score, False Negative Rate (FNR) and False Discovery Rate (FDR) are illustrated. Finally this section explores the comparison between prosed and earlier approaches.

### 4.1 Datasets

#### A. MSRAction3D dataset [36]

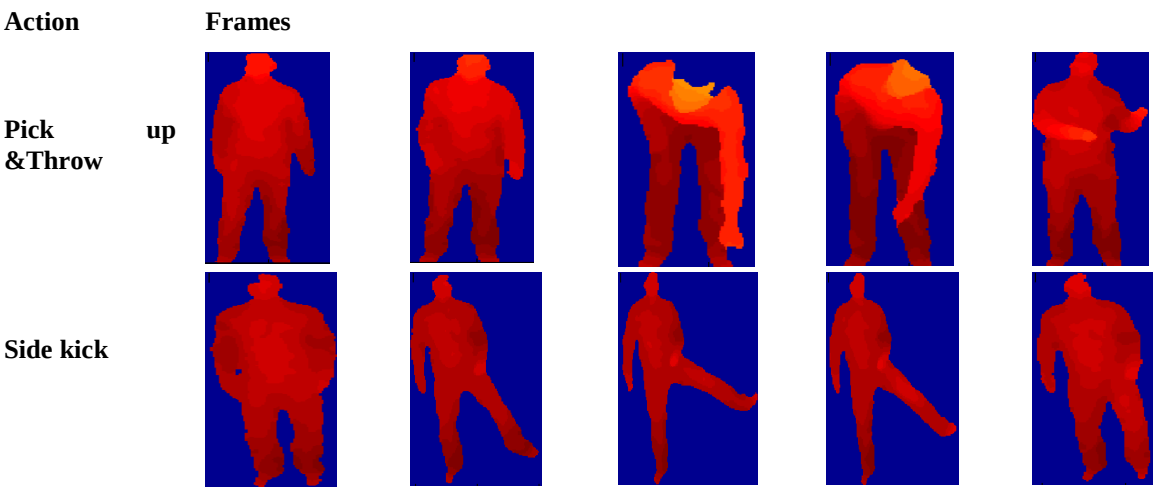
This dataset is of the most popular depth videos dataset and most of the earlier works used this dataset for validation. Totally this dataset consists of 20 different actions: “horizontal arm wave”, “high arm wave”, “high throw”, “forward punch”, “hand catch”, “hammer”, “draw tick”, “draw cross”, “draw circle”, “side boxing”, “two hand wave”, “hand clap”, “jogging”, “side kick”, “forward kick”, “bend”, “pick up & throw”, “golf swing”, and “tennis swing”. A depth camera is used to capture all these actions facing the action with front view. Totally 10 actors are used for this dataset creation and they performed each action 2 to 3 times. Since every actor has his/her own speed, the action videos of this dataset are very challenging. For example the two actions such as Draw tick and Draw cross are much similar and they are differed by just hand movement. Some samples of this dataset are shown in figure.6.

#### B. MSRDailyActivity3D dataset [37]

This dataset is a daily activity dataset consists of totally 16 daily activities: “Call cellphone”, “read book”, “eat”, “drink”, “cheer up”, “use vacuum cleaner”, “use laptop”, “write on paper”, “sit down”, “stand up”, “play guitar”, “walk”, “lay down on sofa”, “play game”, “toss paper”, and “sit still”. Every action is performed under two positions; one is in standby position and another is sitting in sofa position. The total number of depth videos present in this dataset is 320. This dataset contains noises and cluttered backgrounds.

### 4.2 Results

The simulation was done according to the even and odd basis, i.e., for ten subjects of MSR action 3D dataset, the action performed by even number actors are trained and the actions performed by odd number actors are tested. This type of validation is called as cross action validation means the trained and tested actions are not same. Here the actions with actor numbers 1, 3, 5 7 and 9 are trained and the actions with actor numbers 2, 4, 6, 8 and 10 are tested. After testing, the performance is measured through several performance metrics and they are shown in the following table.2 andtable.3.



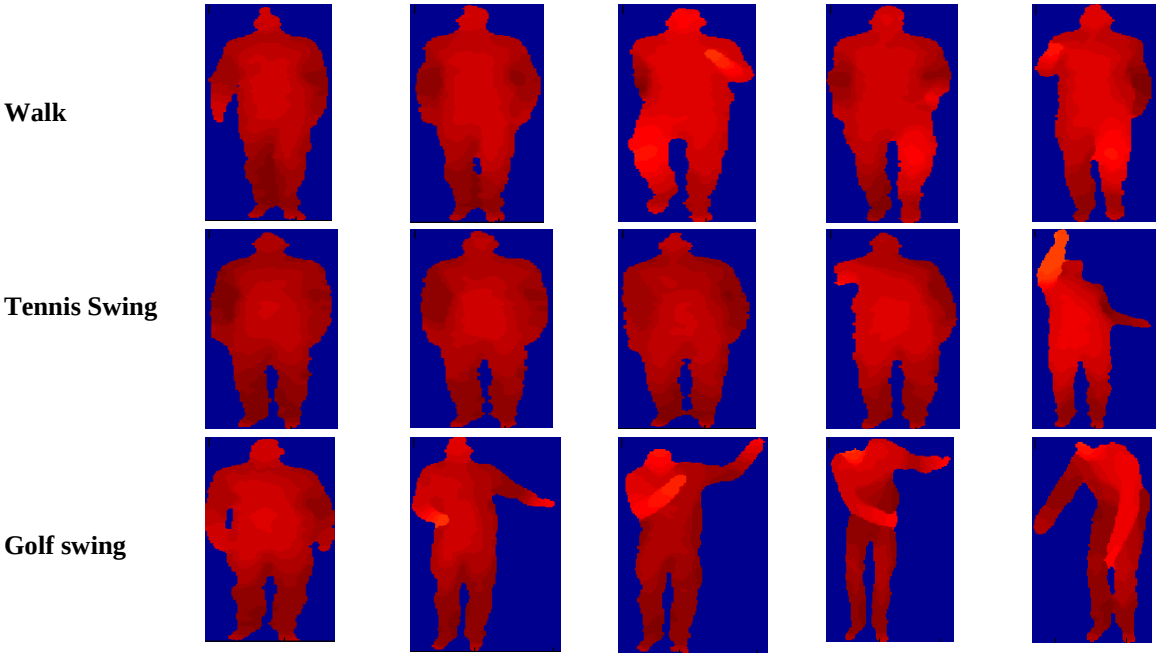
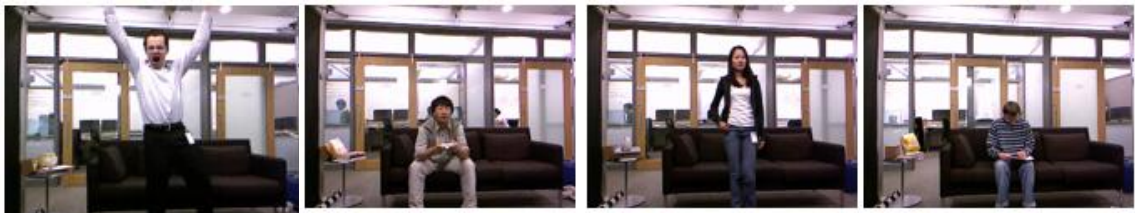



**Figure.6 some samples of MSRAction3D dataset**



(a) (b) (c) (d)

Figure.7 some samples of MSR Daily Activity 3D dataset (a) Cheer up, (b) Play game (c) Stand up, and (d) Read book

Table.2 Performance Metrics of MSR action 3D dataset for different actions

| **Action/Metric** | **TPR (%)** | **PPV (%)** | **F-Score (%)** | **FNR (%)** | **FDR (%)** |
|---|---|---|---|---|---|
| High Arm Wave | 86.0459 | 100.00 | 92.4996 | 13.9541 | 0000 |
| Horizontal Arm Wave | 94.9294 | 86.5292 | 90.5348 | 5.07060 | 13.4708 |
| Hammer | 100.00 | 100.00 | 100.00 | 0000 | 0000 |
| Hand Catch | 79.28922 | 90.9392 | 84.7155 | 20.7107 | 9.0608 |
| Forward Punch | 89.27212 | 100.00 | 94.3320 | 10.7278 | 0000 |
| High Throw | 96.22742 | 92.4931 | 94.3233 | 3.77258 | 7.5069 |
| Draw Cross | 83.29362 | 93.9394 | 88.2967 | 16.7063 | 6.0605 |
| Draw Tick | 95.48432 | 89.0622 | 92.1615 | 4.51568 | 10.9378 |
| Draw Circle | 95.68222 | 100.00 | 97.7934 | 4.31778 | 0000 |
| Hand Clap | 96.59922 | 87.2936 | 91.7109 | 3.40078 | 12.7064 |
| Two-Hand Wave | 100.00 | 94.8575 | 97.2254 | 0000 | 5.1425 |
| Side-boxing | 100.00 | 100.00 | 100.00 | 0000 | 0000 |
| Bend | 87.2547 | 90.2364 | 88.7205 | 12.7452 | 9.7636 |
| Forward Kick | 100.00 | 97.1294 | 98.5437 | 0000 | 2.8706 |
| Side Kick | 100.00 | 93.2215 | 96.8854 | 0000 | 6.7785 |

| | | | | | |
|---|---|---|---|---|---|
| Jogging | 100.00 | 100.00 | 100.00 | 0000 | 0000 |
| Tennis Swing | 95.5336 | 89.0598 | 92.1831 | 4.46639 | 10.9402 |
| Tennis Serve | 95.2725 | 91.4785 | 93.3369 | 4.72750 | 8.5215 |
| Golf Swing | 93.1294 | 97.5992 | 95.3119 | 6.87060 | 2.4008 |
| Pick Up & Throw | 91.5992 | 98.4500 | 95.6154 | 8.40080 | 1.5523 |

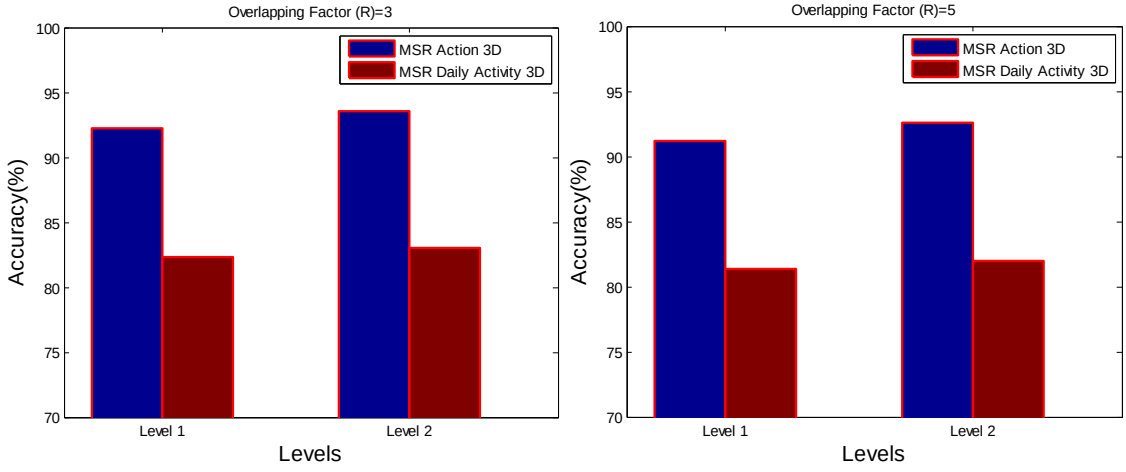
Table.2 shows the details of performance metrics obtained after the simulation of proposed recognition framework over MSR action 3D dataset. The performance evaluation is accomplished totally throng fie metrics such as Recall, PPV, F-score, FNR and FDR. From the results shown in the above table, we have noticed that the maximum recall (100%) is obtained for totally six actions and they are Jogging, Side kick, Forward kick, Two-hand wave, side boxing and Hammer. And the minimum recall is noticed for Hand catch (79.28922%). The next minimum s achieved for Draw cross action (83.29362%). Next, the maximum PPV (100%) is achieved for totally six actions and they are Jogging, side boxing, draw circle, forward punch, hammer and high arm wave. And the minimum PPV is noticed for Horizontal Arm Wave (86.5292%). The next minimum s achieved for Draw Tick action (89.0622%). The main reason behind the les recall rate or less PPV of draw cross and draw circle actions is the structural similarity between them. The three actions namely Draw Tick, Draw cross and Draw circle have almost same movements expect the hand movements and this this main of less PPV. For a given draw cross action frame, the system recognized it as either draw circle or draw tick and hence it has less recall rate. Similarly for a given other inputs like Draw Circle and Draw Cross actions, the system mostly recognized them as draw ticks and hence it has less PPV.

Next, the maximum F-Score (100%) is observed at totally three actions they are hammer, jogging and side boxing. And minimum F-Score is noticed for Hand catch action (84.7155%). The next metrics such as FN and FDR follows an inverse relation with Recall and PPV respectively. Hence for an action which has gained maximum recall will have less FNR and the action which has gained maximum PPV will have less FDR. Based on this we have noticed the minimum FNR (0%) for six actions and they are Jogging, Side kick, Forward kick, Two-hand wave, side boxing and Hammer. And maximum FNR is observed for Hand catch (20.7107%). Similarly, the minimum FDR (0%) is observed for totally six actions and they are Jogging, side boxing, draw circle, forward punch, hammer and high arm wave. And the maximum FDR is noticed for Horizontal Arm Wave (13.4708%).

Table.3 Performance Metrics of MSR daily activity 3D dataset for different actions

| **Action/Metric** | **TPR (%)** | **PPV (%)** | **F-Score (%)** | **FNR (%)** | **FDR (%)** |
|---|---|---|---|---|---|
| Drink | 86.2234 | 71.4458 | 78.1421 | 13.7765 | 28.5542 |
| Eat | 85.4127 | 90.1147 | 87.7007 | 14.5873 | 9.88530 |
| Read book | 95.2252 | 82.6693 | 88.5861 | 4.58480 | 17.3307 |
| Call cellphone | 66.3145 | 65.2471 | 65.7764 | 33.6855 | 34.7529 |
| Write on a paper | 92.1345 | 86.3312 | 89.2226 | 7.68550 | 13.6688 |
| Use laptop | 93.1124 | 75.4214 | 83.3383 | 6.88760 | 24.5786 |
| Use vacuum cleaner | 89.2341 | 92.3312 | 90.8495 | 10.5853 | 7.66880 |
| Cheer up | 76.4147 | 95.4574 | 84.8811 | 23.5853 | 4.54259 |
| Sit still | 96.0032 | 84.4414 | 89.8518 | 3.99679 | 15.5586 |
| Toss paper | 66.1457 | 89.4578 | 76.0554 | 33.8543 | 10.5422 |
| Play game | 50.2345 | 85.3369 | 63.2413 | 49.7655 | 14.6631 |
| Lay down on sofa | 89.4574 | 75.4878 | 81.8810 | 10.5426 | 24.5122 |
| Walk | 85.7741 | 88.6996 | 87.2123 | 14.2259 | 11.3004 |
| Play guitar | 90.3336 | 75.8858 | 82.4817 | 9.66640 | 24.1142 |
| Stand up | 88.8854 | 85.6653 | 87.2456 | 11.1146 | 14.3347 |
| Sit down | 89.4175 | 92.4578 | 90.9122 | 10.5825 | 7.54219 |

Table.3 shows the obtained Performance Metrics after the simulation of proposed recognition framework over MSR daily activity 3D dataset. Towards this simulation, we have considered both set of action depth videos (action videos captured under two positions such as Standby and siting in sofa position) for training and testing. Similar to the above simulation, here also the validation is done through cross subjects, i.e., the subjects of 1, 3, 5, 7, 9 are trained and the subjects of 2, 4, 6, 8, and 10 are tested. In this simulation we have noticed that some actions like read book, use laptop, play guitar and sit still has very much less movements and hence they can be considered as 3D video with static motions. These actions are also processed for simulation and we observed maximum recognition performance. From the table.3, we have noticed that that maximum recall (96.0032%), precision (95.4574%), and F-Score (90.9122%), are obtained for actions sit still, cheer up, and sit down respectively. Next the minimum recall (50.2345%), precision (65.2471%), and F-Score (63.2413%), are obtained for actions Play game, Call cellphone, and play game respectively. Next, the maximum FNR (49.7655%) and FDR (34.7529%) are observed for actions play game and call cell phone respectively. Finally the minimum FNR (3.9967%) and FDR (4.54259%) are observed for actions sit still and cheer up respectively.




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

**Figure.8 Accuracy at different overlapping factors (a) R = 3, and (b) R = 5**

To alleviate the effectiveness of temporal segmentation, we have simulated the proposed action recognition model with varying overlapping factors (R) at both Level 1 and Level. At level, the frames length is considered as 5 and at level 2, the frame length is considered as 10. Initially at both level, we have segmented the frames with overlapping factor R=3 and next with overlapping factor R = 5. After the simulation though these parameters, the obtained accuracy are shown in figure.8, where 8(a) is the accuracy at R = 3 and the 8(b) is the accuracy at R = 5. From these results we have noticed that as the frame length increases, the accuracy also increase but up to certain extent only. For example, the accuracy at level 1 (frame length = 5) for MSR action 3D dataset is observed as (92.2212% whereas it is of 93.5568% at level 2 (frame length = 10).

Similarly the accuracy at level 1 for MSR daily activity 3D dataset is observed as 82.3452% whereas it is of 83.0012% at level 2. Further we have noticed that as the overlapping factor, R value increases, the accuracy reduces. This is illustrates with the following values; the accuracy obtained for MSR action 3D dataset at R=3 and level 1 is observed as 92.2212% whereas it is of 91.2212% at R = 5 at the same level. Similarly the accuracy of MSR action 3D dataset at R=3 and level 2 is observed as 93.5568% whereas it is of 92.1023% at R = 5 at the same level. The similar nature of accuracy results are observed for MSR daily activity 3D dataset also. For instance, the accuracy of MSR daily activity 3D dataset at R=3 and level 1 is observed as 82.3452% whereas it is of 81.3321% at R = 5 at the same level. Similarly the accuracy of MSR daily activity 3D dataset at R=3 and level 2 is observed as 83.0012% whereas it is of 82.3214% at R = 5 at the same level.

### 4.3 Comparison

Table.4 shows the comparison between the proposed and conventional approaches through recognition accuracy on the both MSR action 3D and MSR daily activity 3D datasets. LOP [38] and actionlet [38] are the two typical features which are mainly focused on the encoding of depth data. These methods have limited accuracies, which reflects the challenges (e.g. cluttered backgrounds and noises) of the dataset especially MSR Daily Activity 3D dataset. Using DMM with HOG, X. Yang et al., [8] achieved an accuracy of 88.73% while this approach is not able to capture the body shaking movements due to the simple DMM. Next, combining the

DMM with DMA, Kim D et al., [32] achieved an accuracy of 90.4500% which is more sensitive to the noise and textural variations in the depth action videos. To overcome this problem, Chen et al., [21], [23] employed LBP over the obtained DMM and every action is represented with a fisher kernel vector. Though these methods have gained a better recognition performance, they are not focused over the presence of fake moving pixels in the DMM. These fake moving pixels are derived due to several reasons like small body shaking movements, cluttered backgrounds and low resolution cameras. The conventional DMM cannot capture such kind of noises and hence the proposed approach has better performance compared to them. On an average the recognition accuracy of DMM + LBP methods is noticed as 91.20% while the proposed $D^2$MM+$M^2$HI+CNN is noticed as 93.6410%, shows an improvement of 2.44% in recognition accuracy.

Table.4 Comparative analysis through accuracy

| **Method** | **Dataset** | **Accuracy (%)** |
|---|---|---|
| DTW [16] | MSR Action 3D | 63.8900 |
| DMM + HOG [8] | MSR Action 3D | 88.7300 |
| HON4D [20] | MSR Action 3D | 88.9300 |
| DMA + DMH + HOG [32] | MSR Action 3D | 90.4500 |
| Actionlet Ensemble [38] | MSR Daily Activity 3D | 74.2200 |
| STIPs (Harris3D+HOG3D) [18] | MSR Daily Activity 3D | 60.6000 |
| DMM + LBP [23] | MSR Action 3D | 90.5000 |
| DMM + LBP + FV [21] | MSR Daily Activity 3D | 84.000 |
| DMM + LBP+ LF [21] | MSR Action 3D | 91.9000 |
| LOP Features [38] | MSR Daily Activity 3D | 42.5000 |
| $D^2$MM-CNN [33] | MSR Action 3D | 91.5900 |
| $D^2$MM-CNN | MSR Daily Activity 3D | **85.2330** |
| $M^2$HI-CNN | MSR Action 3D | **90.3620** |
| $M^2$HI-CNN | MSR Daily Activity 3D | **83.3320** |
| $D^2$MM+ $M^2$HI+CNN | MSR Daily Activity 3D | **87.7850** |
| $D^2$MM+ $M^2$HI+CNN | MSR Action 3D | **93.6410** |

Our recent method, i.e., $D^2$MM-CNN [33] has developed an extension to the conventional and DMM obtained an accuracy of 91.5900% after the simulation over MSR Action 3D dataset. Here the same method is applied over the MSR Daily Activity 3D dataset and the recognition accuracy is observe as 85.2330% while for conventional approaches, like STIPs (Harris3D+HOG3D) [18] and Actionlet Ensemble [38], it is of 60.6000% and 74.2200% which is very less. Next, Chen at al., [21] also applied the DMM + LBP+ FV method over MSR Daily Activity 3D dataset and obtained an accuracy of 84.000% which is 1.233% less than the proposed $D^2$MM-CNN.

Next, another proposed method, $M^2$HI-CNN applied over two dataset and gained an accuracy of 90.3620% and 83.3320%. Compared to the conventional approaches, this method also gained a better performance because it has efficiently encoded the motion information by finding the face movements in the human body. Finally the average accuracy of integrated method (i.e., $D^2$MM+ $M^2$HI+CNN) has obtained an accuracy of 93.6410% and 87.7850% after its accomplishment over the MSR Action 3D and MSR Daily Activity 3D datasets respectively, which is high, compared to all the earlier methods. The main reason is that the proposed action representation model is more resilient to the side effects present in the actions videos such as body movements and cluttered backgrounds etc.

## V. CONCLUSION

In this paper, we have proposed a new action descriptor which can identify the real and fake motions in the actions videos and encodes the temporal information between sequences of frames. The proposed descriptor is an integrated form which was constructed by combining depth motion map and motion history information of a depth video. The newly proposed Difference Depth Motion Map can remove the cluttered backgrounds

effectively and can also derive the motion information in the low resolution videos. Further the newly proposed Modified Motion History Image can differentiate the real movements from fake movements and can represent the motion image much effectively. Before the action representation, to include much motion information, a new segmentation called a temporal segmentation s employed at different levels. Simulation experiments are conducted over MSR action 3D dataset and MSR daily activity 3D dataset and the obtained results out-performs the state-of-art methods.

**REFERENCES**

[1] C. Chen, K. Liu, R. Jafari, and N. Kehtarnavaz, "Home-based senior fitness test measurement system using collaborative inertial and depth sensors," *in EMBC*, 2014, pp. 4135–4138.
[2] S. Herath, M.T. Harandi, F. Porikli, Going deeper into action recognition: a survey, *CoRR* abs/1605.04988 (2016)
[3] V. Bloom, D. Makris, and V. Argyriou, "G3d: A gaming action dataset and real time action recognition evaluation framework," *in CVPRW*, 2012, pp. 7–12.
[4] R. Poppe, (2010), A survey on vision-based human action recognition, *Image Vis. Comput.*, 28 (6), 976-990, 2010.
[5] M. Keestra, (2015), Understanding human action: Integrating meanings, mechanisms, causes, and contexts, In Repko Allen, Szostak Rick & Newell William (eds.), *and Interdisciplinary Research: Case Studies of Integrative Understandings of Complex Problems*. Sage Publications. pp. 201-235
[6] Z. Zhang, "Microsoft kinect sensor and its effect,"*Multimedia, IEEE*, vol. 19, no. 2, 2012, pp. 4-10.
[7] L. Chen, H. Wei, and J. Ferryman, "A survey of human motion analysis using depth imagery," *Pattern Recognition Letters*, 2013, pp. 1995-2006.
[8] X. Yang, C. Zhang, and Y. L. Tian, "Recognizing actions using depth motion maps-based histograms of oriented gradients," *ACM MM*, pp. 1057–1060, 2012.
[9] Bobick, A. F., & Davis, J. W. (2001). The recognition of human movement using temporal templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 23* (3), 257–267.
[10] Pisharady, P. K. , & Saerbeck, M. (2015). Recent methods and databases in vision-based hand gesture recognition: A review. *Computer Vision and Image Understanding, 141*, 152–165.
[11] S. Herath, M. Harandi, and F. Porikli, "Going deeper into action recognition: A survey," arXiv preprint arXiv:1605.04988, 2016.
[12] S. Escalera, V. Athitsos, and I. Guyon, "Challenges in multimodal gesture recognition," *Journal of Machine Learning Research*, vol. 17, no. 72, pp. 1–54, 2016.
[13] H. J. Escalante, I. Guyon, V. Athitsos, P. Jangyodsuk, and J. Wan, "Principal motion components for one-shot gesture recognition," *Pattern Analysis and Applications*, pp. 1–16, 2015.
[14] Pham Chinh Huu, Le Quoc Khanh, Le Thanh Ha, "Human Action Recognition Using Dynamic Time Warping and Voting Algorithm", *VNU Journal of Science: Comp. Science & Com. Eng*., Vol. 30, No. 1 (2014) 22-30.
[15] G. D. Forney, "The viterbi algorithm," *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268–278, March 1973.
[16] D. Wu, F. Zhu, and L. Shao, "One shot learning gesture recognition from RGB-D images," in 2012 *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshop*s, June 2012, pp. 7–12.
[17] Y. M. Lui, "Human gesture recognition on product manifolds," *Journal of Machine Learning Research*, vol. 13, no. 1, pp. 3297–3321, Nov 2012.
[18] Klaser, A. , Marszaek, M. , & Schmid, C. (2008). A Spatio-temporal descriptor based on 3d-gradients. In *BMVC 2008-19th british machine vision conference,* (pp. 275–281), September.
[19] Wan, J. , Ruan, Q. , Li, W. , & Deng, S. (2013). One–shot learning gesture recognition from RGB-d data using bag of features. *The Journal of Machine Learning Research, 14* (1), 2549–2582.
[20] Oreifej, O. , & Liu, Z. (2013). HON4d: Histogram of oriented 4d normals for activity recognition from depth sequences. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 716–723) .
[21] Chen, C., Jafari, R. , & Kehtarnavaz, N. (2015). Action recognition from depth sequences using depth motion maps-based local binary patterns. In Proc., of *2015 IEEE winter conference on Applications of computer vision (WACV),* pp. 1092–1099.
[22] Chen, C., Liu, M. , Zhang, B. , Han, J. , Jiang, J. , & Liu, H. (2016). 3d action recognition using multi-temporal depth motion maps and fisher vector. In *IJCAI* (pp. 3331–3337).
[23] Chen, C., Liu, K. , & Kehtarnavaz, N. (2013). Real-time human action recognition based on depth motion maps. *Journal of Real-time Image Processing, 12* (1), 155–163.
[24] Mahmoud Al-Faris, John Chiverton, Yanyan Yang 2 and David Ndzi, "Deep Learning of Fuzzy Weighted Multi-Resolution Depth Motion Maps with Spatial Feature Fusion for Action Recognition", *J. Imaging*2019, 5, 82; doi:10.3390/jimaging5100082.

[25] Jiang Li; Xiaojuan Ban; Guang Yang; Yitong Li; Yu Wang, "Real-time human action recognition using depth motion maps and convolutional neural networks", *International Journal of High Performance Computing and Networking*, 2019 Vol.13 No.3, pp.312 – 320
[26] Xu Weiyao, Wu Muqing, Zhao Min, Liu Yifeng, Lv Bo, and Xia Ting, "Human Action Recognition Using Multilevel Depth Motion Maps", *IEEE Access*, Volume 7, 2019, pp. 41811- 41822.
[27] Wu Li, Q. Wang, and Y. Wang, "Action Recognition Based on Depth Motion Map and Hybrid Classifier", *mathematical problems in engineering*, Vol.2018, Article ID 8780105, 10 pages.
[28] K. Watanabe, T. Kurita, Motion recognition by higher order local auto correlation features of motion history images, *Bio-Inspired, Learning and Intelligent Systems for Security*, 2008. pp. 51–55.
[29] Y.L. Tian, L. Cao, Z. Liu, Z. Zhang, Hierarchical filtered motion for action recognition in crowded videos, *IEEE Trans. Syst. Man Cybern. Part* C 42 (3) (2012) 313–323.
[30] A. A. R Ahad, "Motion History Images for Action Recognition and Understanding", *Springer Book Series*, 2013.
[31] Enqing Chen , Shichao Zhang and Chengwu Liang, "Action Recognition Using Motion History Image and Static History Image-based Local Binary Patterns", *International Journal of Multimedia and Ubiquitous Engineering*, Vol.12, No.1 (2017), pp.203-214.
[32] Kim D, Yun W. H, Yoon H. S, and Jaehong H. S, "Action recognition with depth maps using hog descriptors of multi-view motion," in *proc., of 8th International Conference on Mobile Ubiquitous Computing, Systems, Services, and Technologies, UBICOMM*, pp. 2308–4278, 2014.
[33] S. Sandhya Rani, Dr. G. Appa Rao Naidu, Dr.V. Usha Shree, "$D^2$MM-CNN: Difference Depth Motion Map and Convolutional Neural Networks for Human Action Recognition", *International Journal of Advanced Science and Technology*, Vol. 28, No. 15, (2019), pp. 747-763.
[34] Duin, R. P. (2002). The combining classifier: To train or not to train? In *Pattern recognition, 2002* In *Proceedings. 16th International Conference on: vol. 2* (pp. 765–770).
[35] Wu, D. , Pigou, L. , Kindermans, P. J. , Le, N. D. H. , Shao, L., Dambre, J. , & Odobez, J. M. (2016b). Deep dynamic neural networks for multimodal gesture segmentation and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 38* (8), 1583–1597.
[36] W. Li, Z. Zhang, and Z. Liu, "Action recognition based on a bag of 3d points," in CVPRW, 2010, pp. 9–14.
[37] J. Wang, Z. Liu, Y. Wu, and J. Yuan, "Mining actionlet ensemble for action recognition with depth cameras," in CVPR, 2012, pp. 1290–1297.
[38] J. Wang, Z. Liu, and Y. Wu, "Learning actionlet ensemble for 3D human action recognition," TPAMI, vol. 36, no. 5, pp. 1290–1297, 2014.