Body Joints and Trajectory Guided 3D Deep Convolutional Descriptors for Human Activity Identification

N. Srilakshmi, N. Radha

Abstract*: Human Activity Identification (HAI) in videos is one of the trendiest research fields in the computer visualization. Among various HAI techniques, Joints-pooled 3D-Deep convolutional Descriptors (JDD) have achieved effective performance by learning the body joint and capturing the spatiotemporal characteristics concurrently. However, the time consumption for estimating the locale of body joints by using large-scale dataset and computational cost of skeleton estimation algorithm were high. The recognition accuracy using traditional approaches need to be improved by considering both body joints and trajectory points together. Therefore, the key goal of this work is to improve the recognition accuracy using an optical flow integrated with a two-stream bilinear model, namely Joints and Trajectory-pooled 3D-Deep convolutional Descriptors (JTDD). In this model, an optical flow/trajectory point between video frames is also extracted at the body joint positions as input to the proposed JTDD. For this reason, two-streams of Convolutional 3D network (C3D) multiplied with the bilinear product is used for extracting the features, generating the joint descriptors for video sequences and capturing the spatiotemporal features. Then, the whole network is trained end-to-end based on the two-stream bilinear C3D model to obtain the video descriptors. Further, these video descriptors are classified by linear Support Vector Machine (SVM) to recognize human activities. Based on both body joints and trajectory points, action recognition is achieved efficiently. Finally, the recognition accuracy of the JTDD model and JDD model are compared.

***Index Terms*: HAI, Body joints, Optical flow, JDD, JTDD, C3D, SVM**

I. INTRODUCTION

HAI is the process of recognizing the actions of a person by using the video sequences which contain a complete action execution and retrieving the videos of interest. It can be used in different applications like video surveillance, human-machine interface, smart home [1], healthcare systems [2], etc. In day-by-day, unrealistic numbers of videos are created because of the surveillance systems, movies, YouTube, etc. As a result, HAI becomes an important research area in recent days. Typically, automatic recognition of human abnormal activities in surveillance systems may support the people to aware the related authority of possible illegal or uncertain characteristics. Similarly, the motion recognition in gaming applications can recover the human-machine interface. In healthcare applications, it can support the patient's rehabilitation like automatic recognition of patient's actions can be utilized to facilitate the rehabilitation processes [3-5].

In the earlier period, a number of researches have been projected for different kind of applications based on HAI. In contrast, perfect identification of activities is still a vastly demanding process owing to noisy environments, occlusions, perspective dissimilarities and so on. Most of the recent techniques may provide specific assumptions about the conditions under which the video was captured. But those assumptions were not often employed in real-time applications. Additionally, the two-step approach has been developed in which the attributes from raw video frames were computed and then obtained features were learned by different classifiers. In real-time applications, it was infrequently recognized what features were considerable. In particular, several activity labels may emerge significantly for HAI in terms of their appearances and action models. As a result, different deep learning models have been proposed to train a hierarchy of attributes by constructing high-level attributes from low-level attributes. Those models can be trained to achieve a reasonable performance in HAI systems.

Cao et al. [6] proposed action recognition with JDD to aggregate convolutional activations of a 3D-CNN into discriminative descriptors according to the joint locales. In this method, the video was split into fixed-length clips. For each clip, 3D convolutional feature maps were computed. The annotated or estimated joints of the video were used for localizing points in the 3D feature maps of a convolution layer. Then, the activations at each related locale were pooled and the pooled activations in a similar clip were concatenated together. After that, average pooling and l_2 normalization were utilized for aggregating snip features into video features. Finally, linear SVM was used for the classification process. Moreover, this process was further extended by obtaining the body joint positions [7]. A two-stream bilinear C3D framework was proposed to train the body joints and extract the spatiotemporal features concurrently. After that, the body joint guided feature pooling was achieved by sampling in which the pooling method was devised as a bilinear product function. However, the time consumption for estimating the locales of body joints by using vast data and computational cost of skeleton estimation algorithm were high. Also, the recognition accuracy was needed to be further improvement in an efficient manner.

Revised Manuscript Received on October 10, 2019

N. Srilakshmi, Ph.D Scholar, Department of Computer Science, PSGR Krishnammal College for Women, Coimbatore, Tamilnadu, India.

Dr. N. Radha, Department of Computer Science, PSGR Krishnammal College for Women, Coimbatore, Tamilnadu, India.

Hence, an optical flow extraction is integrated into the two-stream bilinear model efficiently to improve the recognition accuracy. According to this model, optical flows i.e., trajectory points between two video sequences at each body joint positions are extracted automatically. To achieve this, two C3D streams multiplied with the bilinear product is used for extracting the features, generating the pooled descriptors for video sequences and capturing the spatiotemporal features. After that, the whole network is trained for obtaining the video descriptors based on the two-stream bilinear C3D framework that uses the class label. Finally, the linear SVM is used to sort the obtained video descriptors for recognizing human actions. Thus, action recognition performance is improved by considering the optical flow field with body joints efficiently.

II. LITERATURE SURVEY

Ji et al. [8] proposed a 3D-CNN framework in which the spatiotemporal features were extracted by 3D convolutions. Also, the feature from all channels was combined to represent the absolute feature. Then, the framework was regularized by high-level features. Moreover, the outputs of different frameworks were combined for boosting recognition performance. This was evaluated on KTH dataset and the obtained recognition accuracy was 90.2%. Conversely, a number of labelled features were required since it uses a supervised algorithm i.e., gradient-based learning to train this model.

Karpathy et al. [9] proposed a large-scale video classification with CNN to recognize the YouTube videos. In this method, the connectivity of a CNN was extended in a time-domain based on different approaches that take advantage of local spatiotemporal information. The training process was speed-up by suggesting a multi-resolution. The performance analysis was done by using UCF-101 dataset and the recognition accuracy achieved was 65.4%. However, there is a need for improvement on the action recognition. The efficiency can be further improved by combining the clip-level predictions into the global video-level predictions.

Lillo et al. [10] proposed a discriminative hierarchical framework. By using this approach, the human activity classifier was built to simultaneously model which body parts were related to the action of interest with their appearance and composition. Also, when useful annotations were provided at the intermediate semantic level, powerful multiclass discrimination was achieved by learning in a max-margin model. For performance evaluation, two datasets, namely MSR Action3D and CAD120 datasets were used. The recognition accuracy of MSR Action3D and CAD120 dataset was 89.46% and 33.59%, respectively. But, the accuracy of this model was less.

Tompson et al. [11] proposed new hybrid architecture by using a Deep Convolutional Network (ConvNet) and a Markov Random Field (MRF). In this model, a multi-resolution feature representation was used with overlapping fields. Also, this model can approximate MRF loopy belief propagation which was subsequently back-propagated through and learned by using the same learning method as the part-Detector. The recognition accuracy of this model was evaluated on two different datasets such as FLIC and extended-LSP datasets for elbow and wrist joints. For elbow joints, the accuracy of FLIC and extended-LCP datasets was 95% and 66%, respectively. The accuracy for wrist joints using FLIC and extended-LCP datasets were 91% and 62%, correspondingly. However, further improvement of its performance was required.

Cao et al. [12] proposed a spatiotemporal Triangular-chain Conditional Random Field (TriCRF) model for activity recognition. Initially, the difficulty of complex motion identification with an integrated hierarchical framework was addressed. Then, the TriCRF model was expanded to the spatial dimension. In this model, the labels of behaviour were modeled together and their complex dependencies were developed. The accuracy of this model was evaluated on composable activity dataset which was equal to 79%. However, it requires further improvement by incorporating the other layer for learning pose representations jointly with actions and activity.

Wang et al. [13] proposed an action recognition model with the help of Trajectory-pooled Deep-convolutional Descriptor (TDD). In this model, discriminative convolutional feature maps were learned by deep architectures and aggregated into valuable descriptors by trajectory-constrained pooling. As well, two normalization methods such as spatiotemporal normalization and channel normalization were used that transforms convolutional feature maps and enhance the robustness of TDD. The evaluated accuracies for this model using SVM classifier on HMDB51 and UCF101 datasets were 65.9% and 91.5%, respectively. However, body joints were not considered that can help to increase the accuracy efficiently.

Liu et al. [14] proposed an automatic learning of spatiotemporal representation using Genetic Programming (GP) for action recognition. In this model, the spatiotemporal motion features were automatically learned by the motion feature descriptor. The data-adaptive descriptors were learned for various databases with multiple layers and the GP searching space was simultaneously reduced for effectively accelerating the convergence of optimal solutions. The average cross-validation classification error computed by SVM classifier on the training dataset was adopted as the validation measure for GP fitness function. Then, the best-so-far result chosen by GP was obtained as the optimal action descriptor. The accuracy for this model on KTH, YouTube, Hollywood2 and HMDB51 datasets were 95%, 82.3%, 46.8% and 48.4%, respectively. Nevertheless, the processing speed was less.

III. PROPOSED METHODOLOGY

In this part, the JTDD methodology is explained in detail. The two-stream bilinear C3D network framework is applied for automatically predict the spatiotemporal key points in 3D convolutional feature maps with the guidance of body joints with optical flow i.e., trajectory points in the video sequence.

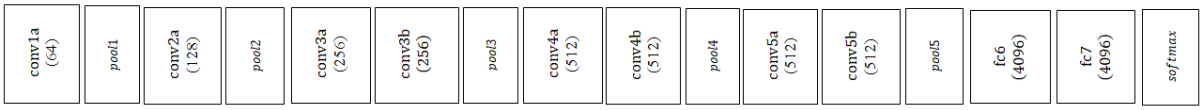

Fig. 1: Architecture of C3D Network

A. Joints and Trajectory-Pooled 3D Deep Convolutional Descriptors

In this process, the C3D network [4] shown in Fig. 1 is used which has architecture as:

$$conv1a(64) - pool1 - conv2a(128) - pool2 - conv3a(256) - conv3b(256) - pool3 - conv4a(512) - conv4b(512) - pool4 - conv5a(512) - conv5b(512) - pool5 - fc6(4096) - fc7(4096) - softmax$$

Here, the number in parenthesis indicates the number of convolutional filters. The number of filters is increased since the combination of the features in each layer is richer than the previous layers. Therefore, increasing number of filters can able to correctly encode the increasingly richer representations of the features. There is a ReLu layer after each convolutional $(conv)$ layer.

Body Joints and Optical Flow Mapping Schemes:

For JTDD, two methods of mapping body joints and optical flow to points in 3D convolutional feature maps, namely fraction scaling and coordinate mapping are compared. Fraction scaling is defined as the fraction of the network's outcome to its input in spatiotemporal-domain for scaling the body joint and optical flow coordinates from the actual video frame into feature maps as follows:

$$(x_c^i, y_c^i, t_c^i) = \left(\overline{(r_x^i \cdot x_v)}, \overline{(r_y^i \cdot y_v)}, \overline{(r_t^i \cdot t_v)}\right) \quad (1)$$

$$(l_c^i, m_c^i, n_c^i) = \left(\overline{(r_l^i \cdot l_v)}, \overline{(r_m^i \cdot m_v)}, \overline{(r_n^i \cdot n_v)}\right) \quad (2)$$

In (1) & (2), $\overline{(\cdot)}$ denotes the rounding operator and (x_c^i, y_c^i, t_c^i) represents the point coordinate in i^{th} 3D convolutional feature maps corresponding to (x_v, y_v, t_v) which is the body joint coordinate in the actual video sequence and (r_x^i, r_y^i, r_t^i) represents the size ratio of i^{th}convolutional feature maps to the video clip in spatial and temporal dimensions. Similarly, (l_c^i, m_c^i, n_c^i) represents the point coordinate in i^{th} 3D convolutional feature maps corresponding to (l_v, m_v, n_v) which is the trajectory point coordinate in the actual video sequence and (r_l^i, r_m^i, r_n^i) represents the fraction of i^{th} convolutional feature maps to the video snip in spatiotemporal-domain.

Coordinate mapping is computing an exact coordinate of the point at the convolutional feature map corresponding to body joint and trajectory point based on the kernel size, stride and padding of each layer. Consider p_i is a point in i^{th} layer, (x_i, y_i, t_i) and (l_i, m_i, n_i) are the coordinate ofp_i. For a givenp_i, the corresponding point p_{i+1} is determined by mapping p_i to the $(i+1)^{th}$ layer. For the convolutional layers and pooling layers, the coordinate mapping from i^{th} layer to $(i+1)^{th}$ layer is developed as follows:

$$x_{i+1} = \frac{1}{s_i^x}\left(x_i + padding_i^x - \frac{k_i^x - 1}{2}\right) \quad (3)$$

$$y_{i+1} = \frac{1}{s_i^y}\left(y_i + padding_i^y - \frac{k_i^y - 1}{2}\right) \quad (4)$$

$$z_{i+1} = \frac{1}{s_i^z}\left(z_i + padding_i^z - \frac{k_i^z - 1}{2}\right) \quad (5)$$

$$l_{i+1} = \frac{1}{s_i^l}\left(l_i + padding_i^l - \frac{k_i^l - 1}{2}\right) \quad (6)$$

$$m_{i+1} = \frac{1}{s_i^m}\left(m_i + padding_i^m - \frac{k_i^m - 1}{2}\right) \quad (7)$$

$$n_{i+1} = \frac{1}{s_i^n}\left(n_i + padding_i^n - \frac{k_i^n - 1}{2}\right) \quad (8)$$

In the above equations, s_i^x, k_i^x and $padding_i^x$ are the $x-$axis element of stride, kernel size and padding of i^{th} layer, correspondingly. Likewise, this is also applied for other dimensions such as y, z, l, m and n, correspondingly.

For ReLU layers, the coordinate mapping correlation is formulated as:

$$(x_{i+1}, y_{i+1}, z_{i+1}) = (x_i, y_i, t_i) \quad (9)$$

$$(l_{i+1}, m_{i+1}, n_{i+1}) = (l_i, m_i, n_i) \quad (10)$$

Once the values of C3D kernel sizes, strides and paddings are applied into (3)-(5) and (9) frequently, the correlation between point coordinates in i^{th} convolutional feature maps and body joint locales in the input video sequence is devised as follows:

$$(x_c^i, y_c^i) = \frac{1}{2^{i-1}} \cdot \left(x_v - \frac{2^{i-1}-1}{2}, y_v - \frac{2^{i-1}-1}{2}\right) \quad (11)$$

$$t_c^i = \frac{1}{2^{i-2}} \cdot \left(t_v - \frac{2^{i-2}-1}{2}\right) \quad (12)$$

Similarly, by applying the values of kernel size, strides and paddings into (6)-(8) and (10) repeatedly, the correlation between point coordinates in i^{th} convolutional feature maps and trajectory points in the input video sequence is devised as follows:

$$(l_c^i, m_c^i) = \frac{1}{2^{i-1}} \cdot \left(l_v - \frac{2^{i-1}-1}{2}, l_v - \frac{2^{i-1}-1}{2}\right) \quad (13)$$

$$n_c^i = \frac{1}{2^{i-2}} \cdot \left(n_v - \frac{2^{i-2}-1}{2}\right) \quad (14)$$

Aggregation of Body Joint Points and Optical Flow:

For classification, the extracted features of frames over time are required to aggregate for obtaining the video descriptor. The positions to pool can be determined by employing body joints and trajectory points in video frames to localize points in 3D feature maps. The pooled representation corresponding to each body joint and trajectory point in a frame of a video sequence is a F dimensional feature vector where F denotes the number of feature map channels. The F dimensional feature vector pooled with the guidance of i^{th} body joint at the t^{th} frame of k^{th} clip is denoted by$f_k^{i,t}$. Similarly, the F dimensional feature vector pooled with the guidance of i^{th} trajectory point at the t^{th} frame of k^{th} clip is represented by$g_k^{i,t}$.

Two methods are used for aggregating the pooled feature vectors in all the frames within a video to a video descriptor. One is fusing all the pooled feature vectors belonging to one frame i.e., a $\mathrm{F} \times N \times O \times L$ dimensional feature as:

$$f_k g_k =$$

$$\begin{bmatrix} f_k^{1,1} g_k^{1,1}, f_k^{2,1} g_k^{2,1}, \dots, f_k^{N,1} g_k^{O,1}, f_k^{1,2} g_k^{1,2}, \\ f_k^{2,2} g_k^{2,2}, \dots, f_k^{N,2} g_k^{O,2}, \dots, f_k^{N,L} g_k^{O,L} \end{bmatrix} \quad (15)$$

In (15), N and O represent the number of body joints and trajectory points in each frame, correspondingly and L denotes the length of the video sequence. After that, average pooling and $L2$ norm are used to fuse k frame representations $\{f_1 g_1, f_2 g_2, \dots, f_k g_k\}$ into a video descriptor where k denotes the number of frames within the video sequence.

Another method of aggregation is fusing the pooled feature vectors corresponding to the body joints and trajectory points in one frame i.e., a $\mathrm{F} \times N \times O$ dimensional feature as:

$$f_k^t g_k^t = \left[f_k^{1,t} g_k^{1,t}, f_k^{2,t} g_k^{2,t}, \dots, f_k^{N,t} g_k^{O,t} \right] \quad (16)$$

After that, one frame is characterized by L representations $\{f_k^1 g_k^1, f_k^2 g_k^2, \dots, f_k^L g_k^L\}$ within the same frame. Max + min pooling is used for aggregating these representations into a frame descriptor.

B. Two-Stream Bilinear C3D Model using Body Joints and Optical Flow

The original body joint and optical flow guided feature pooling in JTDD are realizing by choosing the activations at the corresponding points of body joints and trajectory points on convolutional feature maps. For a given video sequence, a M channel of heat maps $(M = N \times O \times L)$ is generated with the similar spatiotemporal size of the convolutional feature maps to be pooled for each body joint and trajectory point at each frame. In the heat map, the value at the corresponding point of the body joint position and trajectory point is coded as 1, while the others are coded as 0. After that, the process of pooling on one feature map guided by the heat map of one body joint and trajectory point can be formulated as a pixel-wise product between the 3D feature map and the 3D heat map followed by a summation over all the pixels. After that, a two-stream bilinear C3D model is applied to learn the guidance from the body joint positions including trajectory points and capture the spatiotemporal features automatically [4].

Thus, by integrating trajectory points with body joint positions in the two-stream bilinear framework, video descriptors are obtained. Finally, linear SVM [15] is applied for classifying the video descriptors and so the human actions from video sequences are recognized.

Normally, the SVM is built as a hyperplane in an infinite-dimensional space. A perfect HAI is achieved by the hyperplane which has the leading space to the adjacent training sequences of any class i.e., functional margin. The training dataset is represented as a set of instance-label pairs$(x_i, y_i), i = 1, \dots, n, x_i \in \mathbb{R}^n, y_i \in \{-1, +1\}$ where x_i denotes the video descriptors (instances) and y_i denotes the labels. The optimal hyperplane with the maximal margin is achieved by resolving the below unconstrained optimization problem for different classes:

$$\min_w \frac{1}{2} w^T w + \mathrm{F} \sum_{i=1}^{n} \xi(w; x_i, y_i) \quad (17)$$

In (17), $\mathrm{F} > 0$ denotes the penalty parameter and w denotes the weight of training sequencesx_i. By solving this optimization problem, human activities are recognized.

IV. RESULTS AND DISCUSSIONS

In this part, the efficiency of JTDD model is analyzed with the JDD model in terms of recognition accuracy by using Matlab2017b. To evaluate the performance, Penn Action dataset is considered which contains 2326 video sequences of 15 action classes. Here, 50% dataset is taken as the training and the rest 50% is considered as testing dataset. For training an attention model, the dataset is splitting into 1163 training and 1163 testing, randomly. The length of videos is from 50 to 100 frames. The body joint coordinates, trajectory points and C3D features are acted as baselines. Therefore, JTDD with these features is evaluated and compared with different pooling settings. The recognition accuracy is the portion of True Positive (TP) and True Negative (TN) rates among the total number of cases. It is computed as:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (18)$$

In (18),FP is False Positive and FN is the False Negative. The results of body joint and trajectory point extraction are shown in Fig. 2.

Fig. 2(a): Input Video Sequence 1

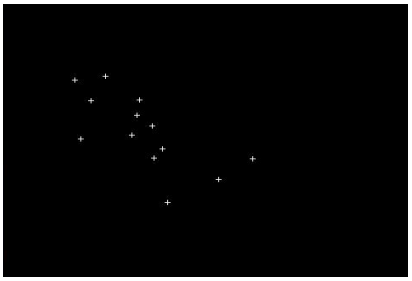

Fig. 2(b): Results for Body Joints Extraction of Input Video Sequence 1

Fig. 2(c): Results for Trajectory Points Extraction of Input Video Sequence 1

Fig. 2(d): Input Video Sequence 2

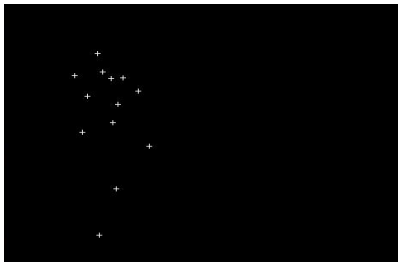

Fig. 2(e): Results for Body Joints Extraction of Input Video Sequence 2

Fig. 2(f): Results for Trajectory Points Extraction of Input Video Sequence 2

In below table, the outcomes on Penn Action dataset are given.

Table 1: Recognition Accuracy of Baselines and JTDD with Various Configurations

	Fuse all the activations	JTDD Fraction Scaling (1×1×1)	JTDD Coordinate Mapping (1×1×1)	JTDD Fraction Scaling (3×3×3)	JTDD Coordinate Mapping (3×3×3)
Joint coordinates + trajectory coordinates	0.6120	-	-	-	-
$fc7$	0.7211	-	-	-	-
$fc6$	0.7368	-	-	-	-
$conv5b$	0.7052	0.8014	0.8599	0.8086	0.8367
$conv5a$	0.6305	0.7583	0.7834	0.7533	0.7628
$conv4b$	0.5324	0.7697	0.7601	0.7847	0.7993
$conv3b$	0.4297	0.7136	0.6845	0.7021	0.7014

From this analysis, it is observed that the accuracy of using the coordinates of body joints and optical flow i.e., trajectory points as a feature is not effective. By using the C3D features which are fusing all the activations of a particular layer as a long vector are highly discriminative because they attain high outcomes. The recognition accuracy of $fc7$ is slightly poorer to that of $fc6$. It is perhaps since the original C3D on Penn Action dataset do not fine-tune that the second fc layer is more fit for the classification of the pre-trained database. For JTDD, the testing on pooling at different 3D $conv$ layers with different body joint and trajectory point mapping formats are publicized.

From Table 1, it is noticed that the JTDD have superior performance compared with C3D features that express the efficiency of body joint and trajectory point guided pooling. The outcomes of various fusing combinations with the scores of SVM on Penn Action dataset is shown in Table 2 and Fig. 3.

Table 2: Recognition Accuracy of Fusing JTDD from Multiple Layers Together with the Scores of SVM

	Fusion Layers					
	$conv5b + fc6$		$conv5b + conv4b$		$conv5b + conv3b$	
	JDD	JTDD	JDD	JTDD	JDD	JTDD
Accuracy	0.855	0.867	0.981	0.987	0.860	0.873

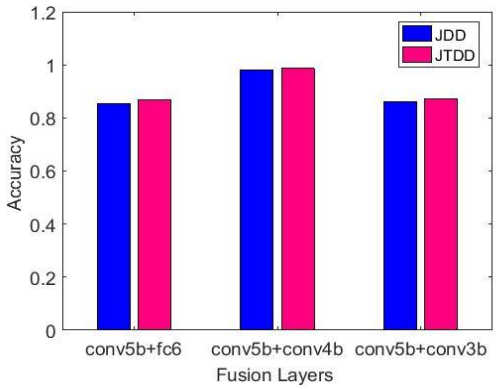

Fig.3: Recognition Accuracy of Fusing JTDD from Multiple Layers

In Fig. 3, it is indicated that fusing JTDDs of different layers certainly increases the recognition outcomes. The combination of JTDDs from $conv5b$ and $conv4b$ increases the recognition performance mostly. High accuracy is achieved by fusing more complementary features.

The results of the impact of estimated body joints + trajectory points versus ground-truth body joints + trajectory points for JDD and JTDD is shown in Table 3 and Fig. 4.

Table 3: Impact of Estimated Body Joints + Trajectories versus Ground-Truth (GT) Body Joints + Trajectories for JDD and JTDD

Method	GT	Estimated	Difference
JDD ($conv5b$)	0.819	0.777	0.042
JTDD ($conv5b$)	0.835	0.810	0.025

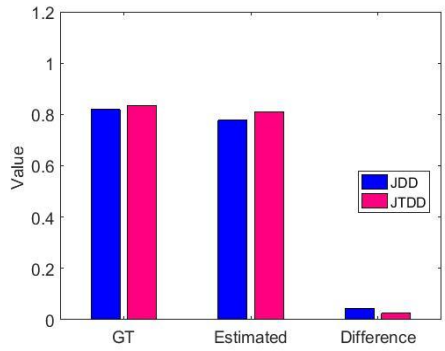

Fig.4: Impact of Estimated Body Joints + Trajectories versus GT Body Joints + Trajectories for JDD and JTDD on Penn Action Dataset

Through Fig. 4, it is noticed that JTDD outperforms competing methods significantly on Penn Action Dataset. JTDD achieves better accuracy not only with GT body joints and trajectory points, but also with estimated body joints and trajectory points, greater than JDD in the order of 10%.

V. CONCLUSION

In this article, JTDD is proposed to extract the optical flow at each body joint positions as the inputs of a C3D model by using two streams of C3D networks which are multiplied with the bilinear product. Based on this, the pooled descriptors for video sequences are generated together and the spatiotemporal features are captured. After that, the entire network is trained end-to-end by using the class label of the two-stream bilinear C3D model to obtain the video descriptors. Moreover, the linear SVM is used to classify the video descriptors for HAR. Finally, the experimental results prove that the recognition accuracy of the proposed JTDD model using Penn Action dataset is increased to 0.987 while fusing JTDDs from $conv5b$ and $conv4b$ with GT body joints and trajectory points. This framework can be applicable for real-time applications such as surveillance, theft identification, motion identification, etc.

REFERENCES

1. A. Sukor, A. Syafiq, A. Zakaria, N. A. Rahim, L. M. Kamarudin, R. Setchi and H. Nishizaki, "A hybrid approach of knowledge-driven and data-driven reasoning for activity recognition in smart homes," J. Intell. Fuzzy Syst., vol. 36, pp. 4177-4188, 2019.
2. J. X. Qiu, H. J. Yoon, P. A. Fearn and G. D. Tourassi, "Deep learning for automated extraction of primary sites from cancer pathology reports," IEEE J. Biomed. Health Inform., vol. 22, no. 1, pp. 244-251, 2018.
3. Y. Kong and Y. Fu, "Human action recognition and prediction: a survey," J. LATEX Cl. Files, vol. 13, no. 9, pp. 1-20, 2018.
4. H. B. Zhang, Y. X. Zhang, B. Zhong, Q. Lei, L. Yang, J. X. Du and D. S. Chen, "A comprehensive survey of vision-based human action recognition methods," Sens., vol. 19, no. 5, p. 1005, 2019.
5. S. R. Ke, H. Thuc, Y. J. Lee, J. N. Hwang, J. H. Yoo and K. H. Choi, "A review on video-based human activity recognition," Comput., vol. 2, no. 2, pp. 88-131, 2013.
6. C. Cao, Y. Zhang, C. Zhang and H. Lu, "Action recognition with joints-pooled 3D deep convolutional descriptors," in Proc.25th Int. Jt. Conf. Artif. Intell., vol. 1, p. 3, 2016.
7. C. Cao, Y. Zhang, C. Zhang and H. Lu, "Body joint guided 3-D deep convolutional descriptors for action recognition," IEEE Trans. Cybern., vol. 48, no. 3, pp. 1095-1108, 2018.
8. S. Ji, W. Xu, M. Yang and K. Yu, "3D convolutional neural networks for human action recognition," IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, no. 1, pp. 221-231, 2013.
9. A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., pp. 1725-1732, 2014.
10. I. Lillo, A. Soto and J. Carlos Niebles, "Discriminative hierarchical modeling of spatio-temporally composable human activities," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., pp. 812-819, 2014.
11. J. J. Tompson, A. Jain, Y. LeCun and C. Bregler, "Joint training of a convolutional network and a graphical model for human pose estimation," in Adv. Neural Inf. Process. Syst., pp. 1799-1807, 2014.
12. C. Cao, Y. Zhang and H. Lu, "Spatio-temporal triangular-chain CRF for activity recognition," in Proc. 23rd ACM Int. Conf. Multimed., pp. 1151-1154, 2015.
13. L. Wang, Y. Qiao and X. Tang, "Action recognition with trajectory-pooled deep-convolutional descriptors," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., pp. 4305-4314, 2015.
14. L. Liu, L. Shao, X. Li and K. Lu, "Learning spatio-temporal representations for action recognition: a genetic programming approach," IEEE Trans. Cybern., vol. 46, no. 1, pp. 158-170, 2016.
15. R. E. Fan, K. W. Chang, C. J. Hsieh, X. R. Wang and C. J. Lin, "LIBLINEAR: A library for large linear classification," J. Mach. Learn. Res., vol. 9, no. Aug, pp. 1871-1874, 2008.

AUTHORS PROFILE

N. Srilakshmi completed MCA and M.Phil degree in Computer Science from Bharathiyar University in the year 2005 and 2008, respectively. Currently, she is working as a guest lecturer of Computer Science in Government Arts College, Udhagamandalam, affiliated by Bharathiyar University. She has 11 years of teaching experience. Her area of interests is data mining and pattern recognition.

Dr. N. Radha, working as an Associative Professor in the department of Computer Science (PG) at PSGR Krishnammal College for Women, Coimbatore. She has 22 years of teaching experience. She has more than 35 publications in both national/international journals. She has guided more than 25 M.Phil scholars. Her research area includes data mining, biometric and information security.