CHAPTER VIII

AN IMPROVEMENTS OF DEEP LEARNER BASED HUMAN ACTIVITY RECOGNITION WITH THE AID OF GRAPH CONVOLUTION FEATURES

One of the most important emerging areas of artificial intelligent is Human Action Recognition (HAR), which automatically detects and labels human actions in video. (Arshad et al. Audio-visual analysis (Hussain et al. 2022), virtual reality (Zhang et al. 2022), intelligent human-machine interactions, etc. (Kulsoom et al. 2022) are only a few examples of the fields where it is very useful. RGB and skeletons are only two examples of the many ways in which human actions might be identified. The skeletal structures embody condensed information about a single motion, which can serve as a robust example of the actions themselves.

Advances in depth sensors will make it easy to record a person's skeletal data, which includes 3D coordinates of major joints. Thus, HAR systems based on a skeleton are currently very popular (Feng et al. 2022). In order to formulate the existence and the temporal dynamics of joints, many conventional skeleton-based HAR approaches manually construct information such the relative placements between joints, the angles between limbs, and the surfaces covered by the human body. Joint coordinates and the strong correlations between them are examples of skeletally based local characteristics. (Cao et al. 2018) argue that this means that methods can't be used to model and distinguish between activities that involve similar stances, movements, and human-machine interfaces. Skeleton prediction is also crucial, because unintended mistakes in locating body joints were not prevented. Video-based HAR uses Convolutional Neural Networks (CNNs) to gather local-to-global features from RGB images and depth data to solve these problems.

With this in mind, the 2-stream Convolutional 3D (C3D) network for HAR was modified to incorporate a Joint and Trajectory-pooled 3D convolutional Descriptor (JTDD) strategy for extracting and merging body joint coordinates and their trajectories. While the C3D network was able to spatially smooth over the neighboring kernels using the highly adaptable max-min pooling, the technique fails to account for the crucial spatial variations between distinct operations. To get around this, the max-min pooling method was abandoned in favor of a Positional Attention-based

Bidirectional Recurrent Neural Network (PABRNN) in the JTDPABRD scheme. However, the BRNN's increased number of parameters makes it less effective in learning long-range joint relations between actions and increases the risk of the gradient vanishing.

Therefore, the PAHBRNN, which segments the feature maps pertaining to the human skeleton in each clip into separate parts based on the body structure, was implemented to construct the JTDPAHBRD scheme. Separate PABRNNs learned these part-based features hierarchically to gather and combine the long-range spatiotemporal data associated to the various body parts. On the other hand, these JTDD variations for HAR relied on the video descriptor, which was built by combining only the joint and trajectory coordinates of various body components at each time step. In order to create more accurate descriptions, it was crucial to determine the geometric relationship between bodily joints. Because joint trajectories do not define contour or geometry relations, they merely express gesture information.

Joints, edges, and surfaces were all taken into account, as were the trajectories of body joints, in order to establish the various forms of geometry using the skeleton graph. The C3D network was fed this information to better understand the temporal dynamics of different geometries through the use of its new View Conversion (VC) layer and Temporal Dropout (TD) layer in the attention and feature streams, respectively. Similarly, to utilized PAHBRNN to get an aggregated representation of features. Then, the bilinear pooling and fully-connected layer were applied to the combined results of the two streams, increasing those results by a factor of two. To classify human actions into their respective buckets, a full network was trained with the softmax loss function to produce frame-specific video descriptors. However, further investigation into the spatial-temporal dynamics of the various skeletal structure geometric aspects was lacking.

The JTDGPAHBRD scheme for HAR is modified in this chapter to include the GCN. As a means of enhancing end-to-end learning and producing video descriptors for a given video sequence, the GCN picks up complementary information between consecutive frames, such as higher-level spatial-temporal properties. To better represent features, a search space is constructed out of many adaptive graph components. The space is then probed using a sampling and computation-efficient

evolution technique. To improve the HAR video descriptor, the GCN's temporal dynamics of the skeleton pattern are combined with the JTDGPAHBRD's geometric features of the skeleton body joints and trajectory coordinates. The SVM technique is then used to classify the collected descriptors in order to recognize various human behaviors. As a result, this approach greatly improves the precision with which various human actions may be identified.

8.1 PROPOSED METHODOLOGY

In-depth information about the GCN model implemented in the JTDGPAHBRD for HAR is provided below. The full research process is depicted in Figure 8.1.

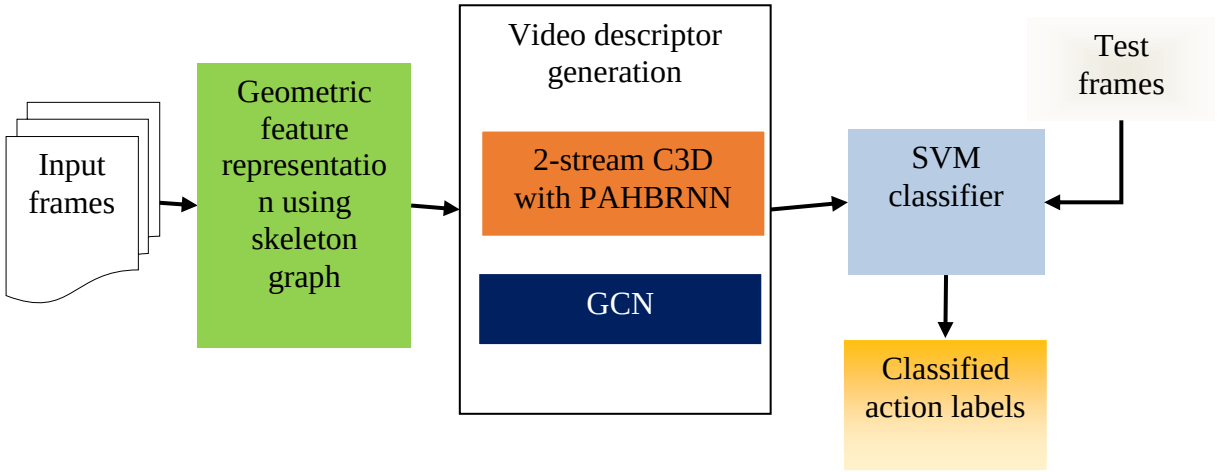


Figure 8.1. Overall Pipeline of the Study

8.1.1 Graph Convolutional Network for Spatial-Temporal Feature Learning

A spatial-temporal graph $G = (V, E)$ with n skeleton geometries and t frames is utilized to represent the skeleton data in the GCN. Therefore, $X \in \mathbb{R}^{n \times t \times c}$ represents the skeletal structure's feature map, where c channels define the joint coordinates.

Typically, an adjacency matrix A and an identity matrix I are used in spatial graph convolutions to define the intra-body joint relations that can be partitioned into three sets s (relative to the group of adjacent resulting from the spatial alignment), with $A + I = \sum_s A_s$. This is the definition of the graph convolution in a particular context:

$$Y = \sum_{i=1}^{s} \Lambda_i^{-\frac{1}{2}} A_i \Lambda_i^{-\frac{1}{2}} X W_i \qquad (8.1)$$

To regularize A_s, let use the degree matrix $\Lambda_s^{ii} = \sum_j (A_s^{ij})$ as shown in Eq. (8.1), and the combined weight vectors for all s, W_i, are shown as well.

Based on the correlations between nodes, this research uses a dynamic and learnable GCN equipped with a search technique to generate dynamic graphs. This standardizes the trainable temporal A over the time-based receptive regions generated by the temporal graph convolutions. This means that the time dispersion between all frames is defined by the S correlation matrix. However, the raw integration of S would result in an inflexible and rigid temporal configuration to all layers, as numerous GCNs are layered to extract high-level spatial-temporal properties. This can be prevented, though, by elevating the geometric features to higher semantic planes. Therefore, the ideal temporal alignment for the hierarchical GCNs is trained using the convolutional layer over the correlation descriptor. The primary result is so described by

$$R_k = ((\mathrm{S}_k W)^\top \mathrm{J}_k)^\top \tag{8.2}$$

Eq. (8.2) represents the temporal dynamics of the similarity matrix with respect to the temporal indices of the kernel patch k, with S_k being the appropriate matrix and J_k being an all-ones vector of dimension c. By adding (8.2) to (8.1) with the Hadamard product, to obtain the temporal descriptor for each feature channel in the graph convolution, allowing for dynamic optimization of the geometric descriptor.

$$Y = \sum_{i=1}^{s} \Lambda_i^{-\frac{1}{2}} A_i \Lambda_i^{-\frac{1}{2}} (X \odot R) W_i \tag{8.3}$$

8.1.2 Search Strategy for Dynamic Graph Generation in GCN

Take a set of Gs, $G = \{G_1, \dots, G_T\}$, where each G defines a skeleton at a certain time interval, then expand them all together. The joints of the skeleton are defined by the nodes and edges of the graph G. The suggested GCN incorporates the graph structure search approach to automatically build graphs for multiple layers at different semantic levels. The search space for a GCN is initially determined, and then many Gs are used in its construction. Then, an exploration policy was presented that minimizes the time and effort spent on sampling and calculation.

GCN search space: What kinds of graph functions an investigating policy could use to build the GCN are defined by the graph search space employed by the graph

structure search method. Here, the optimum GCN for an adaptable G across all levels of interpretation is sought by searching a space constructed from multiple such GCNs. Here are some examples of correlation types used to compute the adaptive G:

1. Topology interpretation relationship: Graph structure is planned using the topological relationship, which takes into account the existing connections between nodes. The degree of similarity between two nodes in a network can be calculated by applying a standard Gaussian function on the network's nodes; this function returns a relationship score.

$$\forall i,j \in V, A_D(i,j) = \frac{e^{\Phi\left(h(x_i)\right)\otimes\psi\left(h\left(x_j\right)\right)}}{\sum_{j=1}^{n} e^{\Phi\left(h(x_i)\right)\otimes\psi\left(h\left(x_j\right)\right)}} \tag{8.4}$$

Spatial m is the name for this element. The interpretations $h(x_i)$ and $h\left(x_j\right)$ of nodes i and j are used to determine the relationship score $A_D(i,j)$. In addition, the channel-wise convolution filters use 2 estimation parameters, denoted by $\Phi(\cdot)$ and $\psi(\cdot)$, and the matrix multiplication, denoted by $\otimes$. From this, they can derive the inter-node correlation necessary to build an adaptive G.

2. Temporal interpretation relationship: Before calculating node associations with Eq. (4), the temporal data of each node is extracted using two temporal convolutions. In this way, the node interfaces between neighboring frames are utilized during the computation of node relations. In addition, the node relationship is computed using a Gaussian function, as shown in Eq. (4). Temporal convolutions apply the functions $\Phi(\cdot)$ and $\psi(\cdot)$ to this element, which is referred to as temporal m.

An adaptive G is constructed with m and t in order to comprehend the spatial and temporal features.

GCN Search Strategy: The optimal graph structure must be investigated in order to lessen the computing complexity of multiple graphs. However, a layer-definite approach is employed to construct a graph because it is claimed that different feature layers contain different depths of semantic information. Thus, the complete GCN network is probed using this computationally efficient method. By making educated guesses about the structural distribution, it locates the best possible layout.

Furthermore, by activating a single function element across all search steps, memory usage is reduced. The structure variables α are treated as a population, and this exploration policy with cross-entropy significance-mixing uses the Gaussian distribution to build the structure distribution. After that, the method selects pivotal examples from a pool of potential structures based on their efficiency, ultimately shifting the overall distribution of those structures. Thus, the optimal structure is drawn at random from the set of all possible structures.

Initially, the structure distribution is modeled with a Gaussian distribution $\pi \sim \mathcal{N}(\mu, \Sigma)$ and N structure examples $S_{new} = \{\alpha_n^i\}_{i=1}^N$ are sampled as the populations for this scheme. After that, combining S_{new} with the past chosen populations $S_{old} = \{\alpha_o^i\}_{i=1}^N$, an importance-mixing scheme is applied to each population to select structure examples. At last, the freshly chosen examples are utilized to modify the structure distribution π.

For each population in S_{old} and S_{new}, the probability density in the pdfs for the π_{new} and π_{old} populations are compared during the population selection process. Thus, in terms of the old population α_o^i,

$$\min\left(1, \frac{p(\alpha_o^i; \pi_{new})}{p(\alpha_o^i; \pi_{old})}\right) > r_1 \tag{8.5}$$

In Eq. (8.5), $p(\cdot; \pi)$ is a pdf with a specific π, and r_1 is a threshold chosen at random between 0 and 1. Similarly, for a new instance α_n^i drawn at random from the current distribution,

$$\max\left(0,1, \frac{p(\alpha_o^i; \pi_{old})}{p(\alpha_o^i; \pi_{new})}\right) > r_2 \tag{8.6}$$

The other threshold in $[0,1]$ is denoted by r_2 in Eq. (8.6). The examples selected in the previous phase are used to adjust the mean μ and covariance Σ for the modification procedure. The model's parameters are adjusted in advance using the existing structure $\alpha = \mu$. The parameters of the network are then set, and the selected examples are assigned to the current architecture. Each sample is evaluated based on how well it performs and ranked accordingly. Each instance i^{th} example is given a significance weight λ_i based on its efficiency rank.

$$\lambda_i = \frac{\log^{(1+N)}/_i}{\sum_{i=1}^{N} \log^{(1+N)}/_i} \tag{8.7}$$

As a result, the distribution can be altered more drastically by giving greater weight to the example with good efficiency. Finally, the structural distribution is adjusted with the help of the weighted examples.

$$\mu_{new} = \sum_{i=1}^{N} \lambda_i \alpha^i \tag{8.8}$$

$$\Sigma_{new} = \sum_{i=1}^{N} \lambda_i \left(\alpha^i - \mu\right)^2 + \epsilon \mathcal{I} \tag{8.9}$$

Eq. (8.9) includes the noise term $\epsilon \mathcal{I}$ for more efficient graph search. Due to the complexity of determining and changing Σ, only a diagonal is allowed. Because the covariance matrix adaption evolution technique shows it to be highly effective, the mean Σ of the last iteration is employed to adjust in Eq. (8.9). Figure 8.2 shows the internal structure of the JTDGPAHBRD-GCN model used to generate video descriptors. Therefore, this GCN using dynamic graphs can represent the spatial-temporal features of the skeletal geometries.

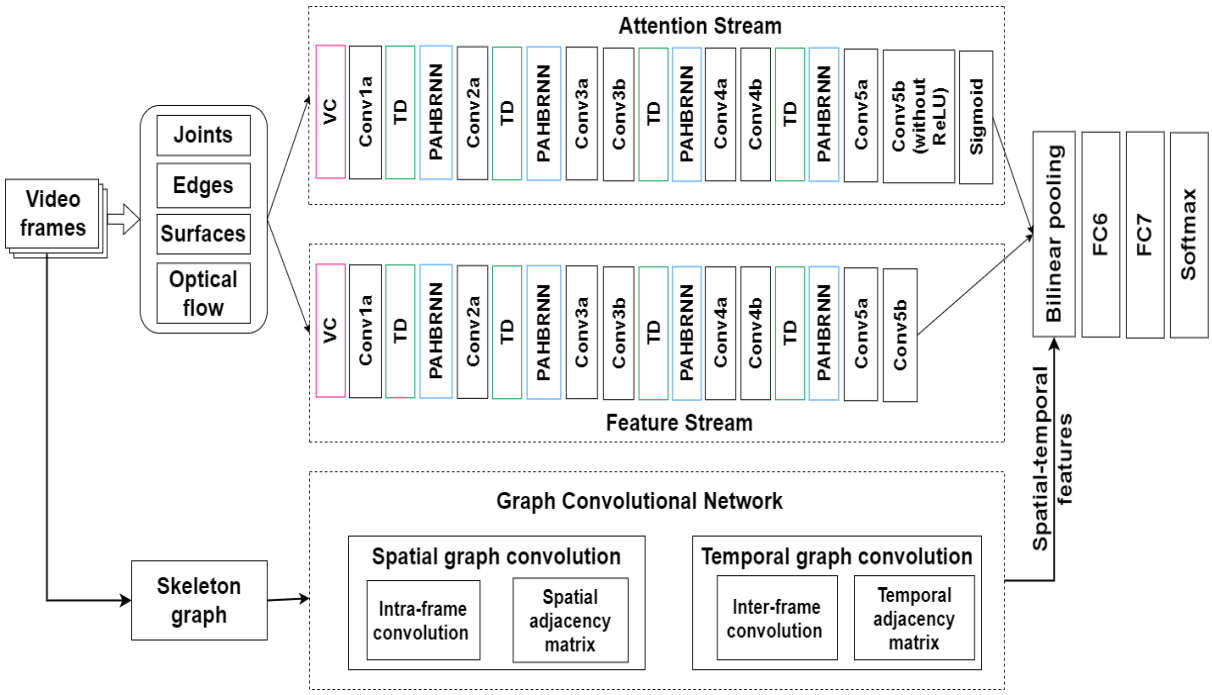


Figure 8.2. Structure of proposed JTDGPAHBRD-GCN Model Video Descriptor Generation

8.1.3 Effective Video Descriptor Generation and Human Action Recognition

The JTDGPAHBRD uses a bilinear product to extract the joints of the body and the coordinates of its trajectory after first retrieving complementing high-level spatial-temporal characteristics from the GCN. To create efficient video descriptors for specific video sequences, the fused feature vectors are then passed to the fully linked layer. At last, the SVM classifier sorts the generated video descriptors into categories of human activities.

8.2 EXPERIMENTAL RESULTS

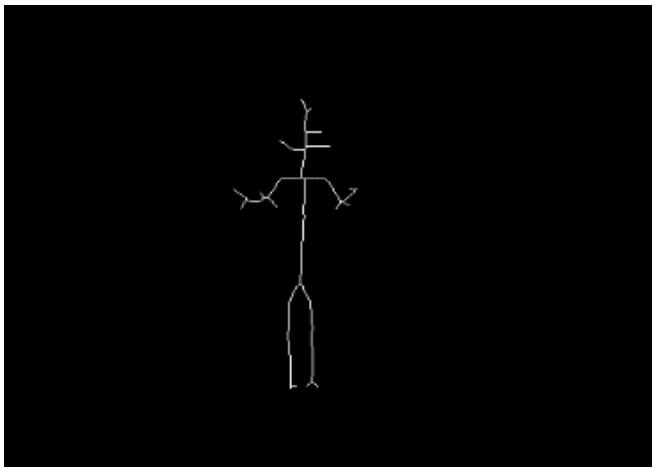
The JTDGPAHBRD-GCN model is tested on a dataset of 2326 video sequences, with each sequence categorized into 15 action classes, using the PAD in MATLAB 2017b. Each clip contains between 50 and 100 blocks, and all of the annotated body joints total 13. There are a total of 1861 training video sequences and 465 test video sequences included in this dataset. Some examples of such data are C3D features, primitive geometry coordinates, trajectory coordinates, and spatial-temporal correlations. Several fusion configurations are used to assess JTDGPAHBRD-GCN's recognition accuracy in light of these features.

Recognition accuracy is the proportion of an individual's action classes that are correctly labeled.

$$Accuracy = \frac{Number\ of\ recognized\ actions}{Total\ number\ of\ actions\ tested} \times 100\% \qquad (8.10)$$

Figure 8.3 depicts the video frame and skeleton image used as input for feature extraction.

Figure 8.3. (a) Input frame



(b)

Figure 8.3. (b) corresponding skeleton image

Table 8.1 displays the results of the JTDGPAHBRD-GCN's recognition accuracy tests on the PAD.

Table 8.1. Recognition Accuracy (%) of Sources and JTDGPAHBRD-GCN with Different Alignments on PAD

	Aggregate all the features	**JTDGPAHBRD-GCN Ratio Scaling (1×1×1)**	**JTDGPAHBRD-GCN Coordinate Mapping (1×1×1)**	**JTDGPAHBRD-GCN Ratio Scaling (3×3×3)**	**JTDGPAHBRD-GCN Coordinate Mapping (3×3×3)**
Geometry features + trajectory coordinates + spatial-temporal features	74.65	-	-	-	-
$\boldsymbol{fc7}$	83.96	-	-	-	-
$\boldsymbol{fc6}$	85.74	-	-	-	-
$\boldsymbol{conv5b}$	82.41	90.11	94.86	89.96	93.08
$\boldsymbol{conv5a}$	73.68	85.35	88.78	84.15	85.44
$\boldsymbol{conv4b}$	65.31	87.19	85.97	88.66	89.23
$\boldsymbol{conv3b}$	54.02	80.54	79.08	80.73	79.67

Table 8.1's first column indicates how well various parameters (joint geometries, trajectory coordinates, and spatial-temporal information) can identify human actions. It finds that human action recognition accuracy is subpar when only a few features are aggregated. As a result, it is necessary to combine all of the features from the various levels to achieve higher precision. When compared to $fc6$, $fc7$ isn't quite as precise. It's promising because the real C3D-GCN can't modify $fc7$, and that's a crucial feature for making a good video description. Results of PAHBRNN-based pooling at each 3D conv units in JTDGPAHBRD-GCN are analyzed because of the incorporation of body joint geometries and trajectory coordinates.

The JTDGPAHBRD-GCN outperformed competing HAR systems on tasks that required it to group together spatial-temporal features in video patterns with the geometries and trajectories of body joints according to different sections of the human body (e.g., the right leg, right arm, trunk, left leg, and left arm).

In addition, JTDGPAHBRD-GCNs from many conv units are averaged to see if they may be balanced. Table 8.2 shows the outcomes of the PAD SVM scores and the outcomes of the late merging setups. There is a comparison made between the JTDPAHBRD-GCN model's accuracy with that of the previously established models (JTDGPAHBRD, TSCN, Hyper-GNN, and Sybio-GNN).

Table 8.2. Recognition Accuracy (%) of JTDGPAHBRD-GCN by Fusing Different Layers for PAD

Concatenation Layers + GCN	**TSCN**	**Hyper-GNN**	**Sybio-GNN**	**JTDGPAHBRD**	**JTDGPAHBRD-GCN**
$conv5b + fc6$	84.36	86.91	88.25	90.60	93.14
$conv5b + conv4b$	94.10	95.39	97.05	99.70	99.82
$conv5b + conv3b$	83.64	85.26	87.48	90.40	92.51

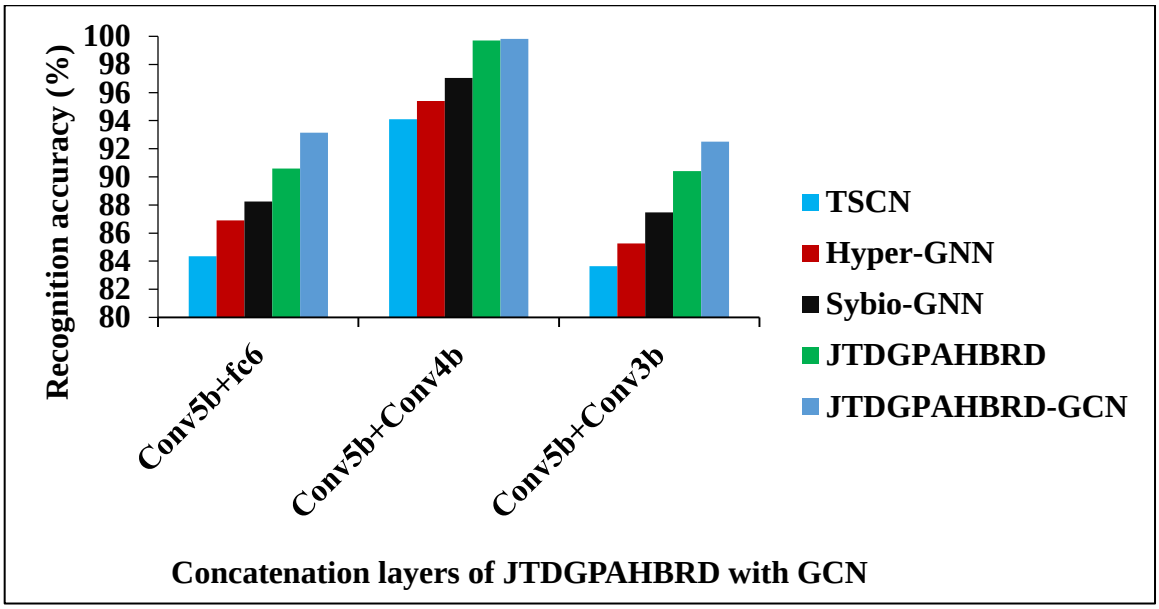


Figure 8.4. Recognition Accuracy of JTDGPAHBRD-GCN on PAD

As can be seen in Figure 8.4, the identification accuracy is highest when combining the $conv5b + conv4b$ features in the JTDGPAHBRD with the GCN features. Therefore, it is concluded that the JTDGPAHBRD-GCN model is superior to previously known models in its ability to accurately categorize human behaviors in constrained video sequences.

Precision, recall, and f-measure for fused layers on the Penn action dataset are shown in Table 8.3.

Table 8.3. Precision, Recall, and F-measure of Fusing Multiple Layers Together on Penn Action Dataset

Performance Metrics	Fusion Layers					
	$conv5b + fc6$		$conv5b + conv4b$		$conv5b + conv3b$	
	JTDG-PAHBRD	JTDG-PAHBRD-GCN	JTDG-PAHBRD	JTDG-PAHBRD-GCN	JTDG-PAHBRD	JTDG-PAHBRD-GCN
Precision	0.885	0.890	0.992	0.995	0.883	0.889
Recall	0.890	0.896	0.996	0.998	0.890	0.896
F-measure	0.888	0.893	0.994	0.997	0.887	0.893

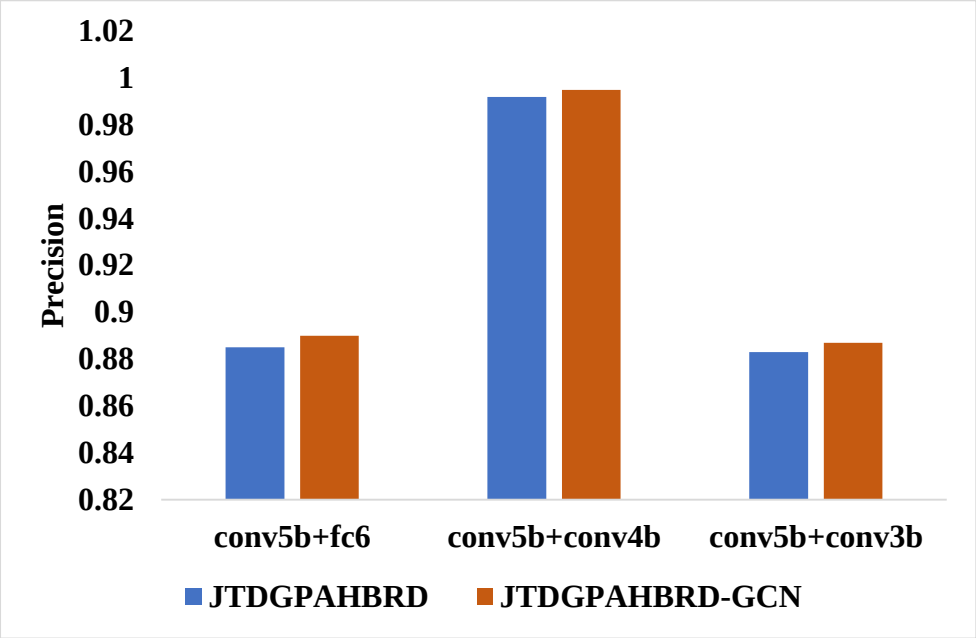


Figure 8.5. Recognition of Precision of JTDGPAHBRD-GCN on PAD

As can be seen in Figure 8.5, the identification precision is highest when combining the $conv5b + conv4b$ features in the JTDGPAHBRD with the GCN features. Therefore, it is concluded that the JTDGPAHBRD-GCN model is superior to previously known models in its ability to accurately categorize human behaviors in constrained video sequences.

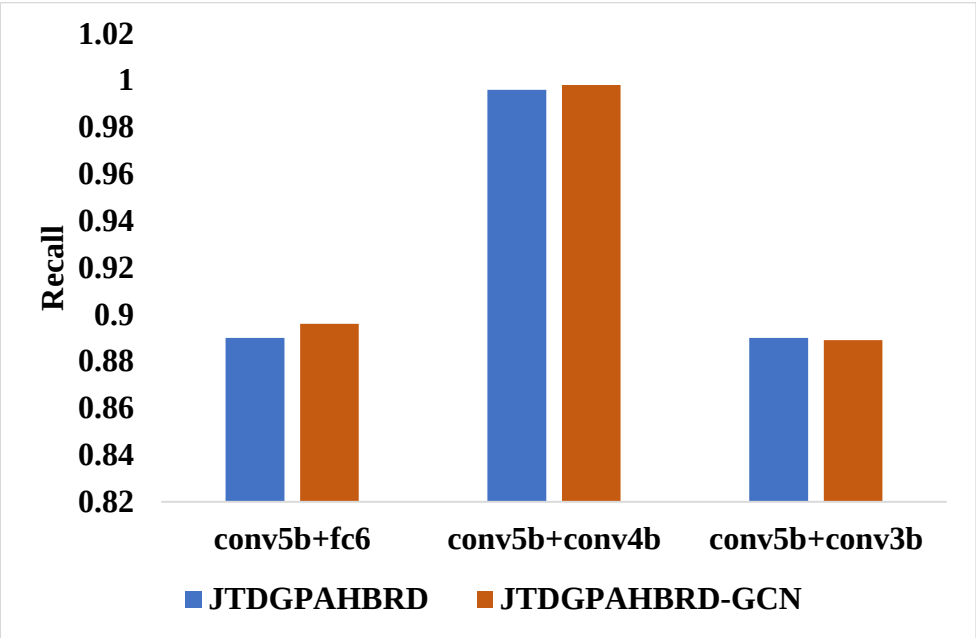


Figure 8.6. Recognition of Recall of JTDGPAHBRD-GCN on PAD

As can be seen in Figure 8.6, the identification recall is highest when combining the $conv5b + conv4b$ features in the JTDGPAHBRD with the GCN features. Therefore, it is concluded that the JTDGPAHBRD-GCN model is superior to previously known models in its ability to accurately categorize human behaviors in constrained video sequences.

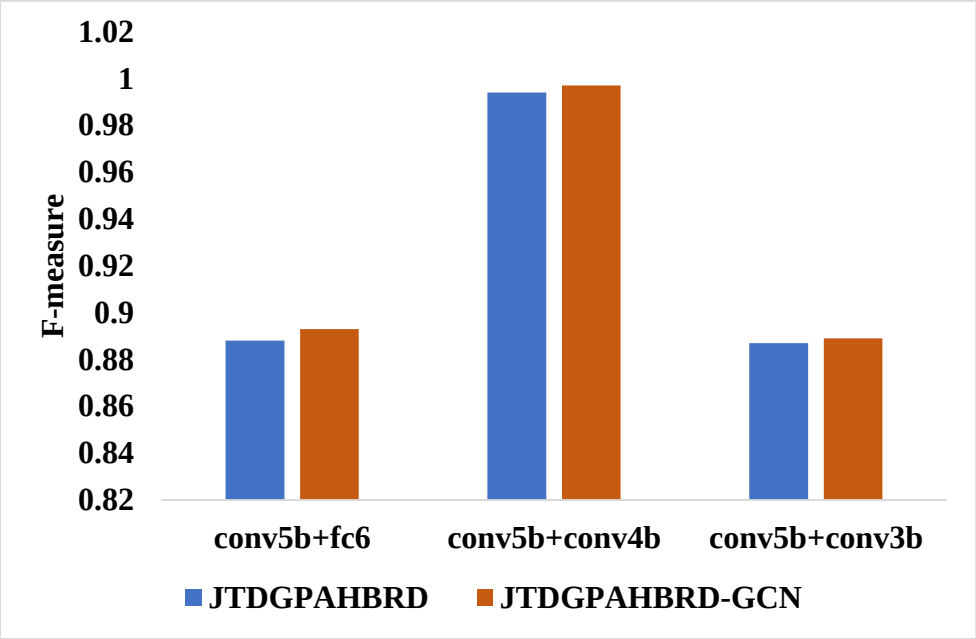


Figure 8.7. Recognition of F-measure of JTDGPAHBRD-GCN on PAD

As can be seen in Figure 8.7, the identification F-measure is highest when combining the $conv5b + conv4b$ features in the JTDGPAHBRD with the GCN features. Therefore, it is concluded that the JTDGPAHBRD-GCN model is superior to previously known models in its ability to accurately categorize human behaviors in constrained video sequences.

Performance results of extracted Geometries+Trajectories+Spatial-Temporal (GTST) features versus ground-truth GTST features for several HAR models on the PAD are provided in Table 8.4.

Table 8.4. Effect of Extracted GTST vs. Ground-truth GTST for Different HAR Models on PAD

Models	Ground-truth	Extracted	Difference
TSCN	0.826	0.801	0.025
Hyper-GNN	0.849	0.833	0.016
Sybio-GNN	0.865	0.851	0.014
JTDGPAHBRD $(conv5b)$	0.893	0.886	0.007
JTDGPAHBRD $(conv5b)$-GCN	0.927	0.922	0.005

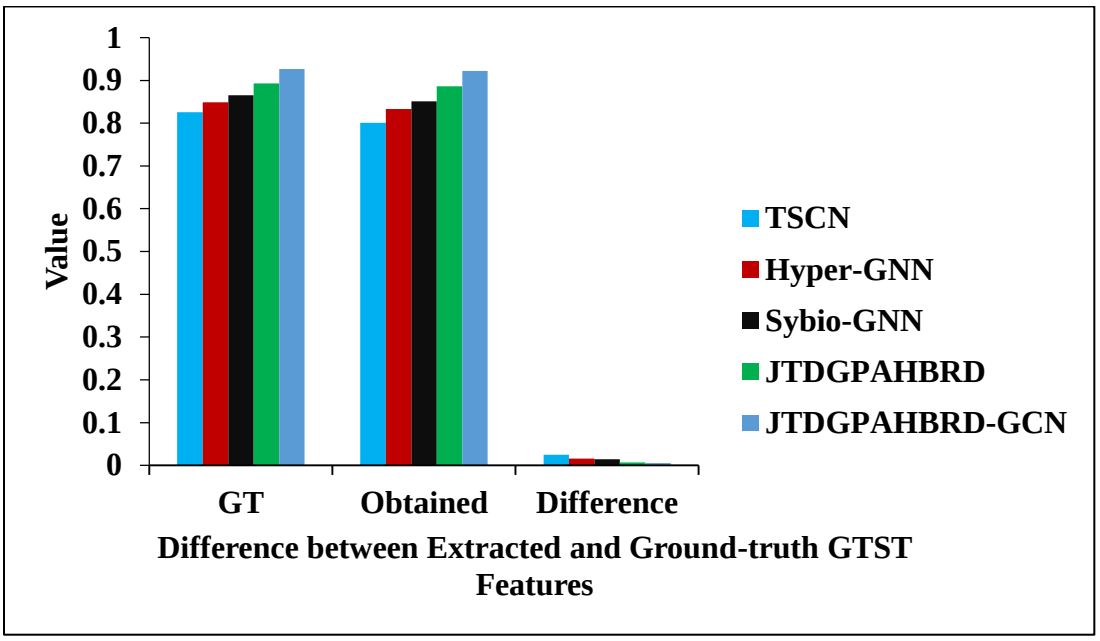


Figure 8.8. Effect of Extracted GTST vs. Ground-truth GTST for Different HAR Models on PAD

As can be shown in Figure 8.8, the JTDGPAHBRD-GCN minimizes the gap between the extracted GTST and the true GTST. After conducting these tests, the PAD found that the proposed JTDGPAHBRD-GCN model had the best recognition performance of all of the HAR models it had tried.

8.3 CHAPTER SUMMARY

This chapter introduces the JTDGPAHBRD, which combines the GCN model with the skeleton graph to learn spatial-temporal information. The GCN was used to capture high-resolution spatial and temporal information between frames. The GCN model's search space, which is made up of many dynamic graph structures, was generated and optimized using a computation-efficient evolution technique so that it could learn the temporal dynamics of the skeletal pattern. The JTDGPAHBRD learned geometric aspects of body joints and their trajectory coordinates, and these were combined with the newly developed spatial-temporal data to provide video descriptors. The SVM classifier for HAR was then used to assign a category to the resulting video description. Finally, the in-depth investigation revealed that, when compared to the other HAR models, the JTDGPAHBRD-GCN model on the PAD achieves a recognition rate of 99.82% through the combination of the features of the $conv5b$ and $conv4b$ layers with the GCN features.