CHAPTER VI

DEEP POSITIONAL ATTENTION-BASED HIERARCHICAL BIDIRECTIONAL RNN WITH CNN-BASED VIDEO DESCRIPTORS FOR HUMAN ACTION RECOGNITION

HAR is a method for obtaining videos that are relevant to a task and identifying a person's unique actions within them. It finds widespread application in fields as diverse as object monitoring, human-computer interface design, and medical aid. Numerous hours of video are captured every day as a result of technologies like surveillance cameras, the web, Livestream, etc. The field of computer vision is likewise increasingly dependent on HAR nowadays (Zhang et al., 2014). Automated identification of certain suspect actions in surveillance systems can accomplish things like automatically identify a person loitering in public places like airports, subway stations, and so on, in addition to helping with the comprehension of inappropriate or irrelevant acts. Different features, such the automatic recognition of many gamers' actions, may become possible with the use of motion recognition. In the medical field, automatic recognition of patient actions can aid in rehabilitation (Ranasinghe et al., 2016).

Low-level HAR, moderate-level HAR, and high-level HAR are the three most common classifications. Edge detection, feature extraction, and action recognition are all carried out in low-level recognition. Human-machine interface identification and deviant behavior recognition are both tasks performed during mid-level recognition. High-level recognitions are also applicable to a variety of complex uses. Different types of HAR systems have been proposed based on the many findings that have been reported over the past few decades. On the other hand, successful action recognition is surprisingly difficult due to factors such as context, individual differences in perception, and so on. Recent methods include video recording in specific scenarios. There has been no implementation of those concepts, however. Additionally, original video streams' attributes are learned and identified in two stages using different classification models. Since feature selection is tough, it is sometimes difficult to identify the features that are crucial for many applications. In particular, the HAR may contain many scenes with completely diverse orientations and paths.

To this end, low- and high-level data extraction has been employed in conjunction with numerous deep learning algorithms for training hierarchical characteristics (Vrigkas et al., 2015). In contrast, action recognition has its own unique set of difficulties due to contextual factors, differences in perspective, and so on. In a number of cutting-edge methods, video is recorded in certain scenarios. However, those concepts have not yet been used in practical settings (Thongrak et al. 2019).

Additionally, several different types of classification models are used in a two-stage learning and identification process to determine what characteristics of the source video streams are being used. Because feature selection is so difficult, many of the qualities that are crucial in many contexts go unnoticed. Specifically, the HAR can contain scenes with wildly varying orientations and trajectories (Basavaiah et al., 2020).

Many deep learning methods, such as feature extraction at varying levels of granularity, have been used to train such hierarchical characteristics (Kim et al., 2019). To ensure sufficient HAR functionality, such techniques are guided by supervised or unsupervised classifiers. Cao et al. (2016) created Joints-pooled 3D-Deep Convolutional Descriptors (JDD) to pool the convolutional activations of the 3D-deep Convolutional Neural Network (3DCNN) into the discriminating descriptors based on the joint coordinates. To begin, they segmented the entire video into smaller, fixed-size pieces and created 3D convolutional attribute maps for each one. Later, the stable joint coordinates were incorporated into the 3D attribute maps of a convolutional unit. They also combined and resampled the activations of each joint coordinate in distinct blocks. These features were then pooled using the mean and l_2-norm to create video descriptors, which were then categorized using a linear support vector machine.

The 2-stream C3D model further developed this strategy by permitting simultaneous joint reference training and spatial-temporal feature extraction. Both preprocessing and skeleton extraction were used to determine the joint coordinates in C3D (Ji et al., 2012). The joint-guided feature vector descriptions of the body were pooled using a max-min method. The resulting video descriptor was computed by feeding the bilinear product of the feature and attention streams into the Fully Connected (FC) layers. However, extracting skeletons was difficult, and finding the joint coordinates was time-consuming for complicated datasets.

Joints and Trajectory-pooled 3D Descriptors (JTDD) have been created (Srilakshmi et al. 2019) to extract and integrate the trajectory coordinates or optical flow between any video streams with the joint coordinates in the C3D approach. The pooled feature descriptors were employed during training, and the generated video descriptor was supplied into the SVM to categorize human activities. Max-min pooling, on the other hand, was used to combine features that are more adaptable to spatial perfection than neighbouring filters. As a result, the necessary differences in location and time across social groups have vanished.

The PA-Bidirectional RNN (PABRNN) model has been incorporated in JTDPABRD (Srilakshmi et al., 2021) to replace max-min pooling for feature aggregation in the two streams of a bilinear C3D network. Combining the body joint and trajectory point coordinates from two independent streams, PABRNN was able to obtain the final video description for HAR. On the other hand, more parameters led to the vanishing gradient issue. In addition, it must take into account previous input sequences in order to properly extract spatiotemporal information from extended video sequences.

In order to better aggregate features, this chapter suggests training a PA-based Hierarchical Bidirectional Recurrent Neural Network (PAHBRNN) on a 3D pool of Joints and Trajectories. Joints and Trajectories Pooled 3D-Deep Positional Attention (PA) based Hierarchical convolutional Recurrent descriptors (JTDPAHBRD) describes this approach. The 2-stream C3D model initially receives the full video pattern in a block-by-block format. After the convolutional layer has recovered the joint and trajectory coordinates, the PAHBRNN is utilized to do feature aggregation instead of max-min pooling. PAHBRNN hierarchically collects the feature vectors corresponding to the different parts of the human skeleton in a given clip using the position-aware guiding vector. Multiplying two streams in a C3D network by their bilinear product during end-to-end training with the softmax loss yields the final video descriptor for a given video sequence. Then, the SVM is fed the video description it has developed in order to identify the person's activities. As a result, it retains sequence information throughout time and is capable of extracting long-term spatiotemporal properties. Back-propagation into the past does not typically result in its disappearance either. This

effectively increases the accuracy with which human actions can be recognized in video sequences.

6.1 PROPOSED METHODOLOGY

An abbreviated explanation of the JTDPAHBRD method is provided below. To begin, each video pattern is split up into many segments and then supplied into the 2-stream C3D model. The attention and feature streams, via the convolutional layer, mine the joint and trajectory coordinates and the spatiotemporal aspects of different human skeleton sections in each clip. After that, data from all of the channels is combined to reveal how activations at different joints and along different trajectories, as well as their associated spatiotemporal properties, affect different body parts. Instead of using max-min pooling, PAHBRNN is used for this purpose. PAHBRNN analyzes five different skeleton-related feature vectors: left arm (LR) and left leg (LL), trunk (TK), right arm (RA) and right leg (RL), and head (HB). These, along with the deep features, are first taken from the C3D model's convolutional layer. The collected characteristics are then sent into five separate PABRNNs after the convolution layer. Motions from nearby skeletal features are generated by combining the interpretation of the trunk feature with those of four other feature types. Then, the locations of features in all the videos that belong to the sequence are compiled. A position-aware guiding vector is assigned to each extracted feature vector related to the human skeleton, and this method is then utilized to propagate the feature vectors to all other positions. For this reason, the position-aware guidance vector offers distinct vectors for each skeletal area. In addition, the resulting aggregated feature vectors are calculated by multiplying each individual feature vector by its appropriate attention weight. Therefore, features can be extracted and the dimensionality reduced by the CNN with PAHBRNN.

Also, the full human skeleton's position is automatically retrieved as a guidance-based feature vector that is then utilized to train a bilinear C3D network. Multiplying the two streams by the bilinear product. Using softmax loss and class labels as supervision, the complete network is trained. So, in order to determine the character of the person's behaviors, the SVM extracts the feature descriptor for a given video sequence and classes it. The JTDPAHBRD-based HAR is depicted in Figure 6.1, while the 2-stream C3D built with PAHBRNN is depicted in Figure 6.2.

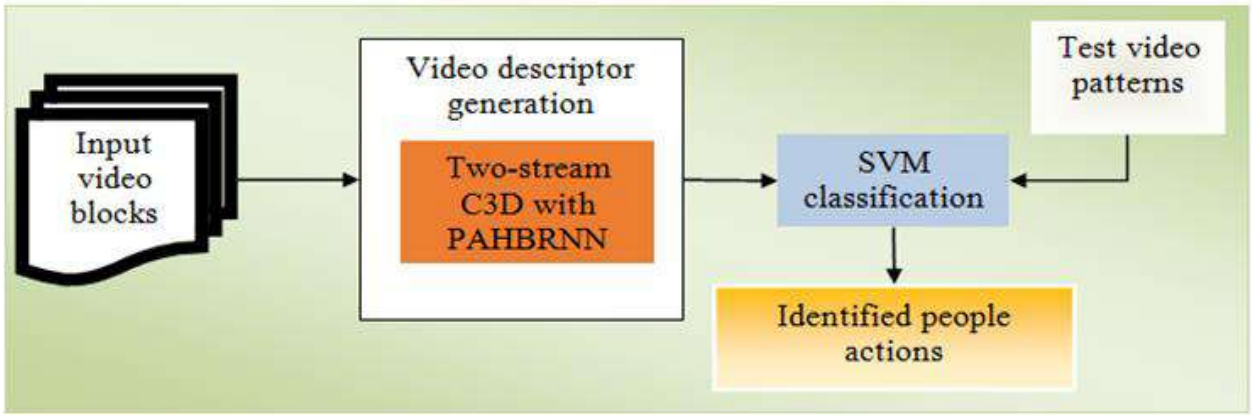

Figure 6.1. Schematic representation of JTDPAHBRD-based HAR

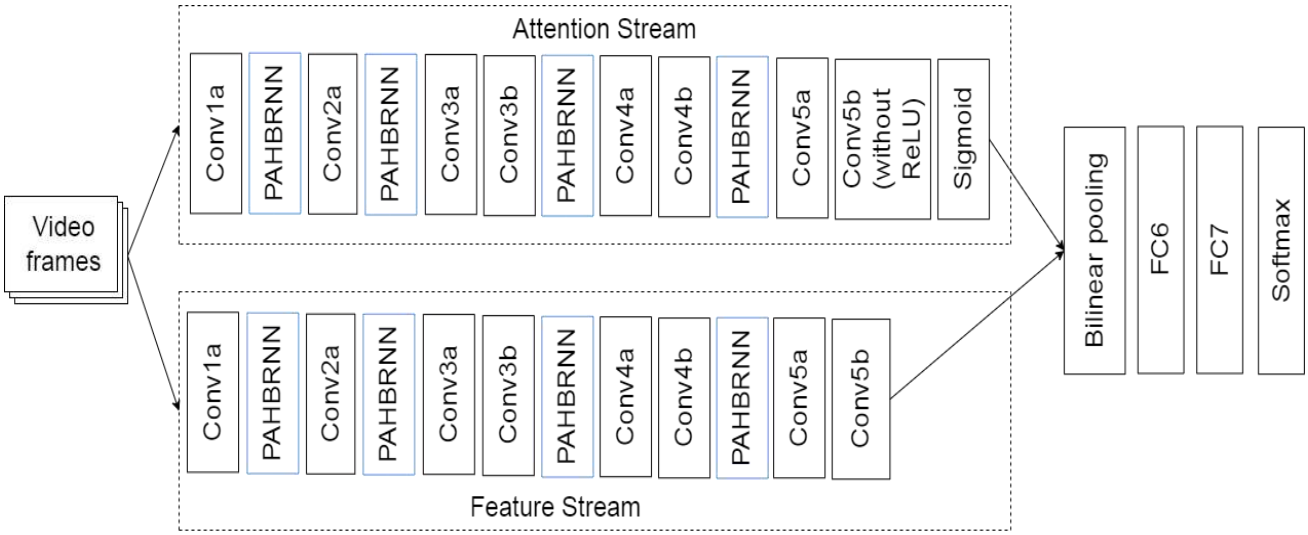

Figure 6.2. Architecture of 2-stream bilinear C3D with PAHBRNN-based feature aggregation approach

6.1.1. Positional attention-based Hierarchical BRNN

Only a subset of humans can perform even the most basic of activities. For example, hitting and kicking forwards require only a slight tilt of the arms and legs. Changing the position of the upper or lower body is rarely necessary for accomplishing most tasks. In addition, the coordinated actions of these 5 body parts create extraordinarily complex activities; for example, both jogging and sailing require the use of complex body movements.

In order to correctly identify a wide range of human activities, it is essential to model the motions of such segments and their combinations. This is why they introduce

the PAHBRNN to extract spatiotemporal patterns from long-term context data. The human skeleton is depicted in Figure 6.3 as a PAHBRNN feature vector representation.

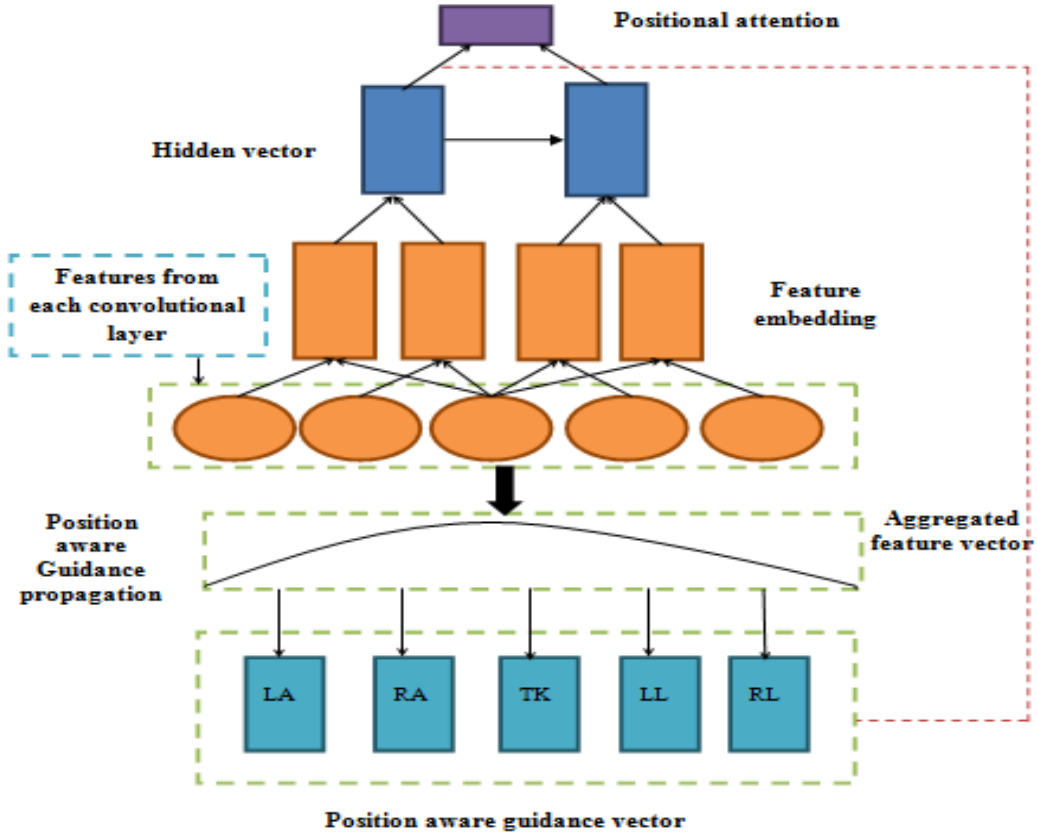

Figure 6.3. Aggregated feature vector representation for entire human skeleton using PAHBRNN model

Taking into account the many feature embeddings (body joints and trajectory points derived from various regions of the human skeleton, including the LA, RA, TK, LL, and RL), PAHBRNN employs the HBRNN with positional attention technique to represent the feature vectors. Keep in mind that the Gaussian kernel is used to direct successive video clips based on positional advice included in the features.

$$Kernel(d) = e^{\left(-d^2/2\sigma^2\right)} \quad (6.1)$$

Where d is the dissimilarity between the true and averaged feature vectors and σ is the variable that determines the extent of the propagation. Given a distance d and an initial point i, it may be define the guidance base matrix G as follows:

$$G(i,d) \sim N(Kernel(d), \sigma') \quad (6.2)$$

For a given estimate of $Kernel(d)$ and standard deviation σ', the mean density is defined by N. A feature's position-specific guidance vector (i.e. LA, RA, TK, LL, and RL) is likewise generated by summarizing the guidance of all features collected from the videos.

$$A_j = Gc_j \tag{6.3}$$

Where

$$c_j(d) = \sum_{f \in F}[(j-d) \in pos(f)] + [(j+d) \in pos(f)] \tag{6.4}$$

The number of features at different distances can be estimated with the help of the distance count vector c_j, which is implemented in Eqns. (6.3) and (6.4), where A_j represents the summed direction vector for the feature located at coordinate j. As an added bonus, let f denote a body joint location or a trajectory point feature in F, let $pos(f)$ denote the set of f's occurrence positions throughout all clips, and let $[\cdot]$ denote an indication function equal to 1 if the criteria satisfy; else equal to 0.

Additionally, the attentive weight (α_j) of the aggregated feature at the location j is incorporated with the position-aware guiding vector for the specific feature as:

$$F_a = \sum_{j=1}^{l} \alpha_j h_j \tag{6.5}$$

Where

$$\alpha_j = \frac{e^{\left(e\left(h_j, A_j\right)\right)}}{\sum_{k=1}^{l} e^{\left(e\left(h_k, A_k\right)\right)}} \tag{6.6}$$

$$e\left(h_j, A_j\right) = v^T tanh\left(W_H h_j + W_A A_j + b\right) \tag{6.7}$$

Eqns. (6.5) and (6.6) utilize the video series length, l, the aggregated feature vector F_a for the full human skeleton in a given clip, the hidden vector h_j at position j, and the aggregated position-aware guidance vector A_j from Eq. (6.3). The matrices W_H and W_A, the bias vector b, the hyperbolic tangent function tanh, the global vector v, and its transpose v^T are all used in Eq. (6.7). The hidden vector and the position-aware guiding vector are used in the score function $e(\cdot)$, which determines the importance of the features.

To properly exploit the diverse components of the human skeleton, this PAHBRNN employs five individual PABRNNs to generate feature vectors. In addition, the bilinear product of the two streams in C3D is used to calculate the final video descriptor, which is then combined with the aggregated feature vectors for all blocks (Vrigkas et al., 2015). To learn to recognize human behaviors, the video descriptors are fed into a support vector machine (SVM).

Algorithm:

Input: Training video patterns

Output: Human actions

Begin

Split video sequences into blocks;

$\boldsymbol{for}(each\ frame)$

Set CNN variables for attention and feature streams;

Extract the features from different parts of the entire human skeleton such as LA, RA, TK, LL and RL at convolutional layers;

Concatenate the features extracted from each convolutional layer using PAHBRNN;

//PAHBRNN

Create the position-aware guidance propagation through Gaussian filter using Eq. (6.1);

Compute $G(i,d)$ by Eq. (6.2);

Aggregate the guidance of (i) RA and LA, (ii) RL and LL with TK features;

Aggregate the guidance of the upper and lower body to get the resultant aggregated position-aware guidance vector using Eqns. (6.3) (6.4);

Get the final combined feature vector belonging to a human skeleton in a single clip by calculating α_j and $e(h_j, A_j)$ using Eqns. (6.5), (6.6) & (6.7);

Fuse attention and feature streams in C3D network with the aid of bilinear product;

Train the two-stream C3D network end-to-end using softmax loss for a whole video sequence;

Obtain the final video descriptors;

Apply the SVM classification;

Identify the human actions from a specified video;

$\boldsymbol{end\ for}$

End

6.2 EXPERIMENTAL RESULTS

This section uses a MATLAB 2017b implementation of the JTDPAHBRD method to examine its performance on the Penn Action dataset. There are a total of 2326 video clips with 15 different labels in this dataset. Annotations for 50–100 blocks, comprising 13 body joints per block, are applied to each video that has been culled from various web video archives. There are a total of 1861 training video sequences and 465 testing video sequences in this trial. C3D characteristics, together with the joint and trajectory coordinates, are also taken into account. Different configurations of feature aggregation are investigated in order to determine how they affect JTDPAHBRD's recognition accuracy.

Accuracy refers to the percentage of accurately identified human actions.

$$Accuracy = \frac{No.of\ recognized\ actions}{Total\ no.of\ actions\ tested} \times 100\% \quad (6.8)$$

Figure 6.4 depicts the experimental outcomes of joint extraction and trajectory coordinate extraction.

Figure 6.4. (a) Sample input video image

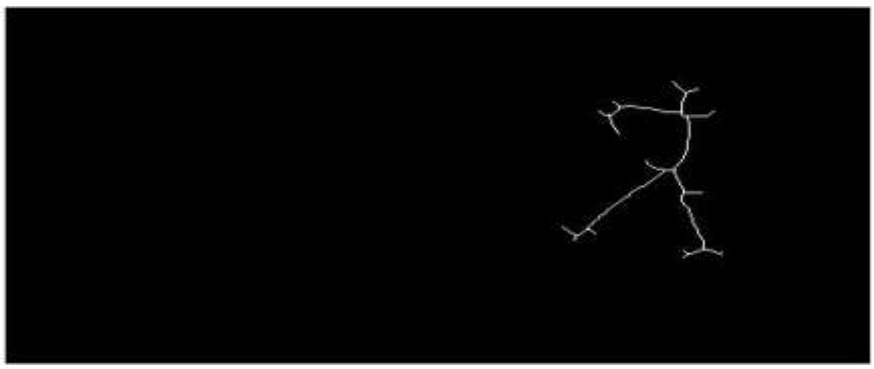

Figure 6.4. (b) Outcomes of joint and trajectory coordinate extraction

Table 6.1 shows the results of JTDPAHBRD's recognition accuracy tests on the Penn Action dataset.

Table 6.1. Recognition accuracy (%) of sources and JTDPAHBRD with different settings on Penn action dataset.

	Aggregate all the activations	**JTDPAHBRD Ratio Scaling (1×1×1)**	**JTDPAHBRD Coordinate Mapping (1×1×1)**	**JTDPAHBRD Ratio Scaling (3×3×3)**	**JTDPAHBRD Coordinate Mapping (3×3×3)**
Joint + trajectory coordinates	0.6621	-	-	-	-
$FC7$	0.7758	-	-	-	-
$FC6$	0.7983	-	-	-	-
$conv5b$	0.7605	0.8533	0.9064	0.8542	0.8885
$conv5a$	0.6834	0.7956	0.8257	0.7961	0.8032
$conv4b$	0.5817	0.8134	0.8015	0.8385	0.8471
$conv3b$	0.4826	0.7517	0.7293	0.7554	0.7566

Joint and trajectory coordinate recognition accuracy, including C3D characteristics, are shown in the first column of Table 6.1. The feature of directly recognizing joint and trajectory coordinates shows a lack of appropriate accuracy. In order to maximize efficiency, it is necessary for all the features in a given layer to aggregate. $FC7$ is marginally less precise than $FC6$. Since the true C3D can't tweak the $FC7$ codec needed to make a good video description, this is doable. Results of PAHBRNN-based pooling at different 3D $conv$ units in JTDPAHBRD are analyzed to account for the larger number of joints and trajectory coordinates. The JTDPAHBRD beats the JTDPABRD, JTDD, and JDD when training on a video pattern comprised of 5 individual segments using only the directed feature vectors of joint and trajectory coordinates.

Additionally, JTDPAHBRDs from various $conv$ units are added together to see whether or not they can achieve balance. Table 6.2 displays the outcomes of late fusion using SVM scores on the Penn Action dataset. Existing methods including JDD,

STDDCN, DWnet, CorNet, JTDD, and JTDPABRD are compared against JTDPAHBRD's accuracy.

Table 6.2. Recognition accuracy (%) of aggregating JTDPAHBRDs from different units on penn action dataset

Aggregation Layers	JDD	STDDCN	Dwnet	CorNet	JTDD	JTDPABRD	JTDPAHBRD
$conv5b + FC6$	85.5	85.8	86.1	86.3	86.7	88.3	88.9
$conv5b + conv4b$	98.1	98.2	98.4	98.5	98.7	99.4	99.6
$conv5b + conv3b$	86.0	86.2	86.5	86.8	87.3	88.3	88.6

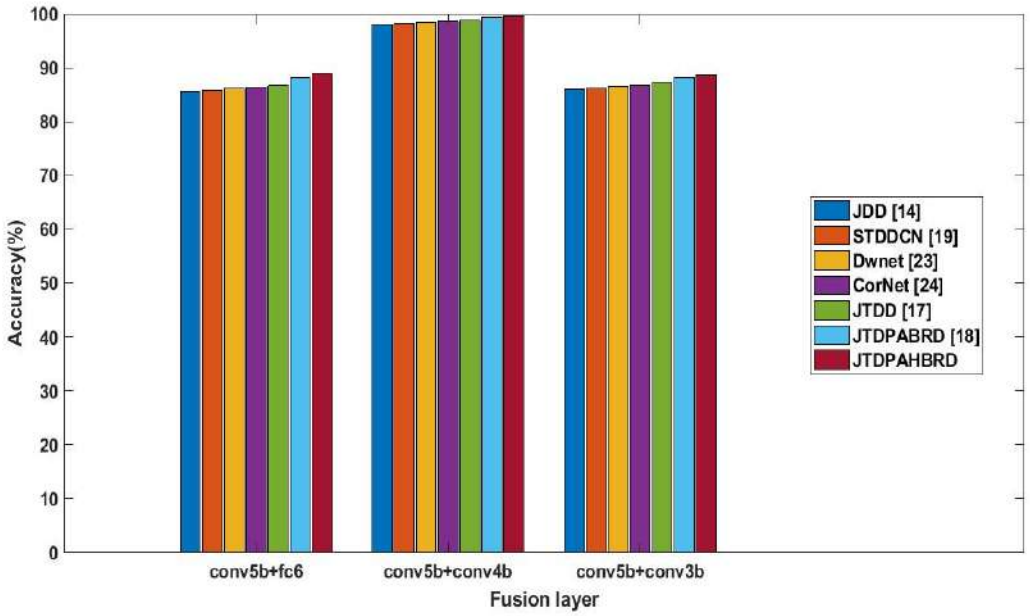

Figure 6.5. Accuracy of aggregating JTDPAHBRD from different units on Penn action dataset

Figure 6.5 depicts the interrelated nature of the characteristics, and evidence suggests that $conv5b + conv4b$ provides the highest accuracy when fusing JTDPAHBRD. Therefore, the JTDPAHBRD method successfully increases the accuracy of all previous methods for recognizing human actions from video sequences.

Fusing many layers together on the Penn action dataset yielded the following values for precision, recall, and f-measure:

Table 6.3. Precision, Recall, and F-measure of Fusing Multiple Layers Together on Penn Action Dataset

Performance Metrics	Fusion Layers					
	$conv5b + fc6$		$conv5b + conv4b$		$conv5b + conv3b$	
	JTDPABRD	JTDPAHBRD	JTDPABRD	JTDPAHBRD	JTDPABRD	JTDPAHBRD
Precision	0.874	0.881	0.983	0.989	0.871	0.879
Recall	0.880	0.886	0.991	0.994	0.878	0.885
F-measure	0.877	0.884	0.987	0.992	0.875	0.882

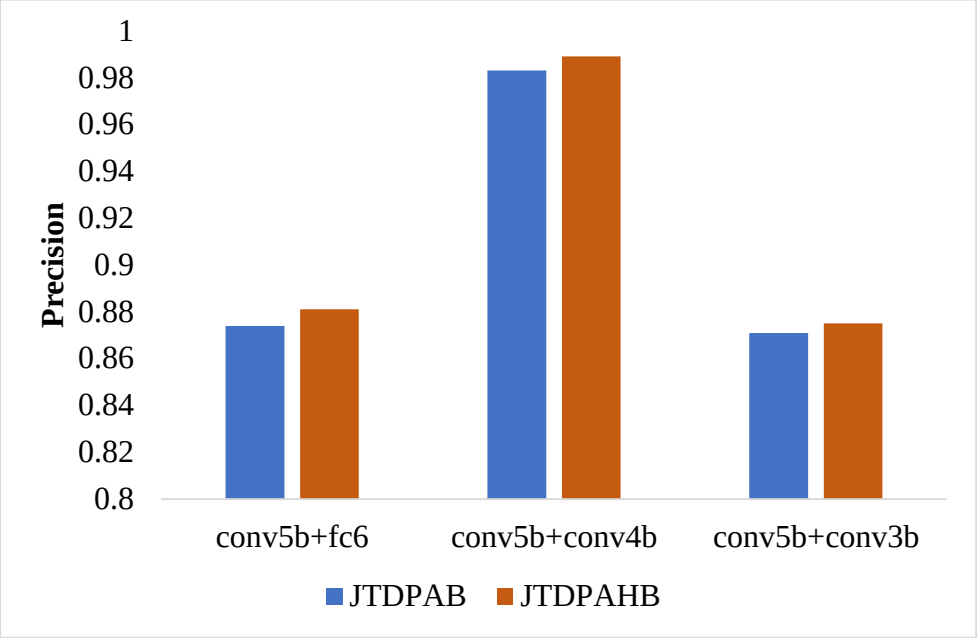

Figure 6.6 Precision of aggregating JTDPAHBRD from different units on Penn action dataset

The features are interconnected, as shown in Figure 6.6, and the precision of fusing JTDPAHBRD from $conv5b + conv4b$ is greater than that of other combinations. Therefore, the JTDPAHBRD method successfully increases the precision of all previous methods for recognizing human actions from video sequences.

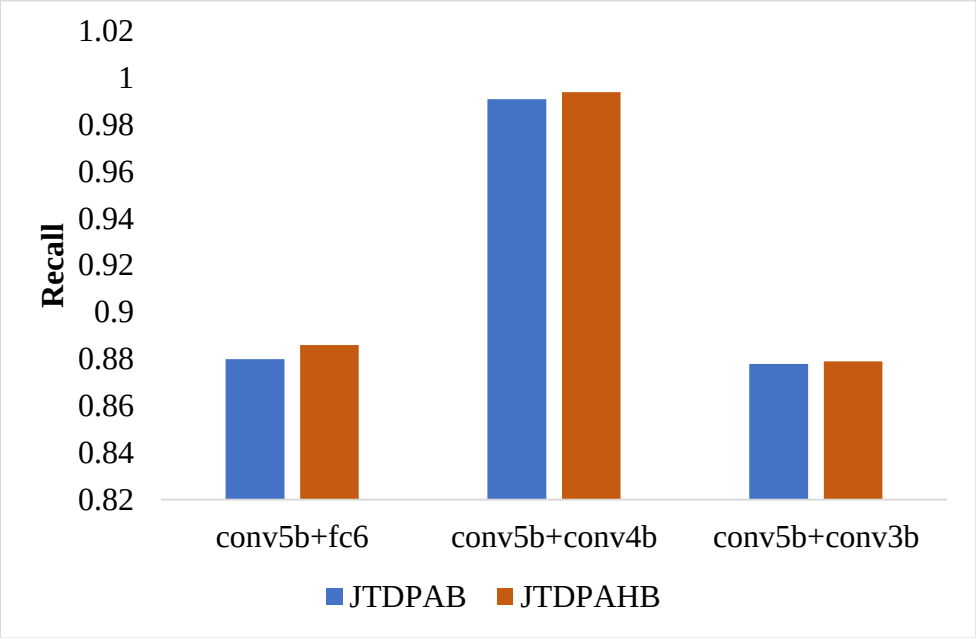

Figure 6.7 Recall of aggregating JTDPAHBRD from different units on Penn action dataset

The features are interconnected, as shown in Figure 6.7, and the recall of fusing JTDPAHBRD from $conv5b + conv4b$ is greater than that of other combinations. Therefore, the JTDPAHBRD method successfully increases the recall of all previous methods for recognizing human actions from video sequences.

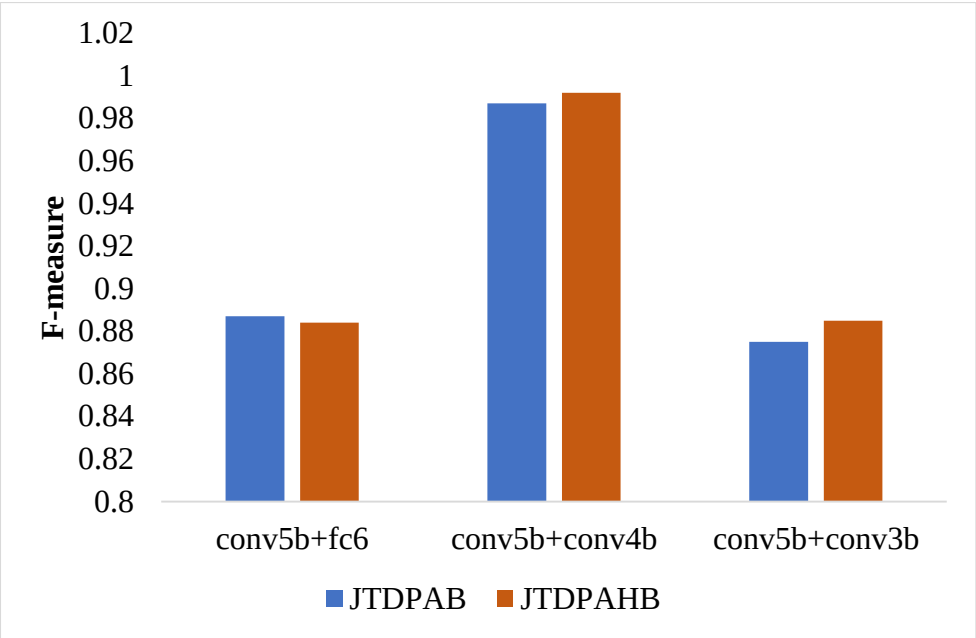

Figure 6.8 F-measure of aggregating JTDPAHBRD from different units on Penn action dataset

The features are interconnected, as shown in Figure 6.8, and the recall of fusing JTDPAHBRD from $conv5b + conv4b$ is greater than that of other combinations. As a result, the F-measure of all prior approaches for identifying human movements from video sequences is significantly improved by the JTDPAHBRD approach.

Table 6.4 compares the impact of extracted joints + trajectory coordinates to Ground-Truth (GT) joints + trajectory coordinates, as well as the results of proposed and existing HAR approaches aggregated from $conv5b$ on the Penn Action dataset.

Table 6.4.Effect of extracted joints + trajectories vs. GT joints + trajectories for proposed and existing approaches on Penn action dataset

JTDPABRD

Approaches Pooled from $conv5b$	**GT**	**Extracted**	**Variance**
JTDPABRD	0.847	0.828	0.019
JTDPAHBRD	0.860	0.849	0.011

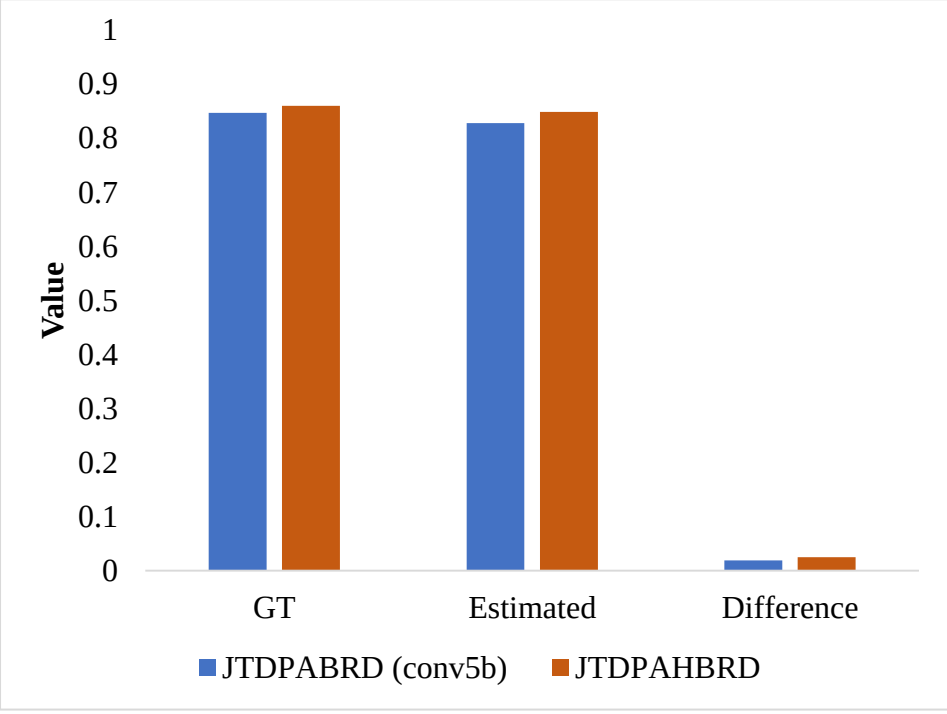

Figure 6.9. Influence of identified joints + trajectories vs. GT joints + trajectories for various approaches on Penn action dataset

As shown in Figure 6.9, the JTDPAHBRD successfully reduces the discrepancy between the GT joints+ trajectory coordinates and the derived joints+ trajectory coordinates to an extremely small value. Comparing its results to those of its peers, such as the JDD, STDDCN, DWnet, CorNet, JTDD, and JTDPABRD, it is clear that it achieves better results on the Penn Action dataset.

6.3 CHAPTER SUMMARY

In this research, PAHBRNN is used to propose the JTDPAHBRD technique, which aggregates features from many video sequences. To begin, the human skeleton is broken down into its component parts and supplied into a 2-stream C3D model. When the PAHBRNN receives this information, it uses a hierarchical structure to combine all of the features into a single vector. To top it all off, the softmax loss is employed throughout the C3D network's training to obtain the final video descriptor. The gathered video description is then used to train a support vector machine (SVM) to recognize the person's actions. The study's findings concluded that JTDPAHBRD coupled with $conv5b$ and $conv4b$ outperformed all other methods on the Penn Action dataset by 1.07%, improving accuracy to 99.6%. Using a mixture of $conv5b + FC6$ layers, the JTDPAHBRD outperforms the competition on the Penn Action dataset with an accuracy of 88.9%. On the Penn Action dataset, the JTDPAHBRD outperforms all other methods by 2.01% thanks to the use of a combination of $conv5b + conv3b$ layers, resulting in an accuracy of 88.6%.