CHAPTER V

DEEP POSITIONAL ATTENTION-BASED BIDIRECTIONAL RNN WITH 3D CONVOLUTIONAL VIDEO DESCRIPTORS FOR ACTION RECOGNITION

HAR is the method used to identify an individual's actions in videos by identifying those that include a particular activity and retrieving those recordings. Industries such as video processing, healthcare IT, and user interface design may have profited from this. The constant stream of new videos from surveillance equipment, the news, YouTube, and other sources is staggering. As a result, HAR has great significance in modern machine learning. Automatic recording of an individual loitering with bags at airports, train stations, and other public places requires us to be able to understand potentially inappropriate or ambiguous behavior on the part of the relevant authority. As an added advantage, the automated detection of several players' trajectories in a given scenario is only one example of how the identification of trajectories can improve the user experience while interacting with computers. In the medical field, this could be used to aid in the recovery of patients by automatically recognizing patient behaviors (Wan et al., 2020).

There are often three levels of representation used in HAR: the lowest, the intermediate, and the highest. Edge detection, feature extraction, interpretation, and action identification are the three primary operations carried out by the low-level representation. First, a single entity is divided into a series of videos, and then characteristics such as texture, attitude, silhouettes, motion, etc. are extracted. The motion detection classifier is trained with these characteristics to identify certain actions. According to Ding et al. In the same vein, the mid-level representation employs three crucial techniques: specific HAR, detection of user-machine interaction, and detection of abnormal behavior. Over time, several high-level representation (HAR) applications are implemented. Numerous studies for various HAR systems have been proposed during the past few years.

On the flip side, it is extremely challenging to accurately identify behaviors due to factors such as context complexity, individual differences in perspective, and other such factors. Most cutting-edge techniques include recording video under controlled conditions. However, real-time systems have not adopted these ideas just yet. The properties of raw video sequences are learned by utilizing various classifiers, and their functions are computed in a two-stage approach. Because the choice of features is so problem-specific in real-time systems, it is difficult to determine which ones are most important. To be more specific, it's possible that there is little to no visual or behavioral distinction between many common HAR behaviors. As a result, deep learning methods have been designed specifically for the purpose of learning hierarchical features by extrapolating generalizations from specific examples (Nweke et al., 2018). Both supervised and unsupervised algorithms are used in their training to produce satisfactory results in HAR.

Cao et al. (2017) propose Joints-pooled 3D-Deep convolutional Descriptors (JDD), which are more effective than other deep learning methods by pooling the convolutional activations of a 3D-deep Convolutional Neural Network (3DCNN) to create discriminative descriptors based on joint locations. The videos is first cut into segments of uniform length, after which 3D convolutional feature maps are calculated for each segment. The 3D feature maps of a convolutional layer are used to pinpoint the predictable joint positions. After that, all the joint location activations from the same clips are added together. In addition, the clip features are aggregated into video features using the average pooling and l_2-norm. Once the features have been labeled, a linear SVM is employed to do so. For better guiding learning from the joints and simultaneous spatiotemporal feature extraction, this technique is improved as the two-stream bilinear C3D network. Manual annotation or skeleton estimate are used to determine joint locations in C3D (Ji et al., 2012). In addition, the vector representations of the body joints are aggregated via max-min pooling. These two streams are multiplied by the bilinear product function and supplied to the fully linked layers to be used as a video description. However, with large datasets, joint location estimate takes more time, and skeleton location estimation is more challenging.

For this reason, Srilakshmi et al. (2019) propose using JTDD, which incorporates an optical flow into the C3D approach. The trajectory points or optical flow between two video sequences can be automatically retrieved at the junction sites by multiplying two C3D streams (including feature and attention) with a bilinear product function. When extracting spatiotemporal features, pooling descriptors are also created. The video descriptors are then trained using the class labels over the entire network. Furthermore, the SVM is utilized to classify these video characteristics for the purpose of locating particular behaviors. However, a type of feature aggregation called max-min pooling was used because of its ability to spatially smooth out differences between neighboring kernels while maintaining its adaptability. This does away with the need for class labels to vary in space and time.

The PABRNN model is incorporated into a two-stream C3D network in this chapter, and Joints and Trajectory-pooled 3D-Deep Positional Attention-based Bidirectional Recurrent convolutional Descriptors (JTDPABRD) are proposed for extracting the important spatiotemporal features and enhancing the accuracy of recognizing individual activities. The original video is broken down into smaller segments that are fed into the two-stream C3D network as input. Joint orientation is extracted from the attention stream and important spatiotemporal aspects of the trajectory are extracted from the feature stream in a C3D network. After that, the convolutional feature vector representations of each clip in the video are aggregated using the PABRNN to generate the clip descriptor. Also, the bilinear product of these two streams is employed together with class labels to train the entire pipeline. In addition, the activations of completely connected layers and the differences between them in space and time are combined to form the ultimate video description. To identify specific actions inside videos, the SVM is fed this video description. This effectively improves the HAR system's accuracy.

5.1 PROPOSED METHODOLOGY

The JTDPABRD approach is briefly discussed here. Figure 5.1 shows a block schematic of this JTDPABRD procedure.

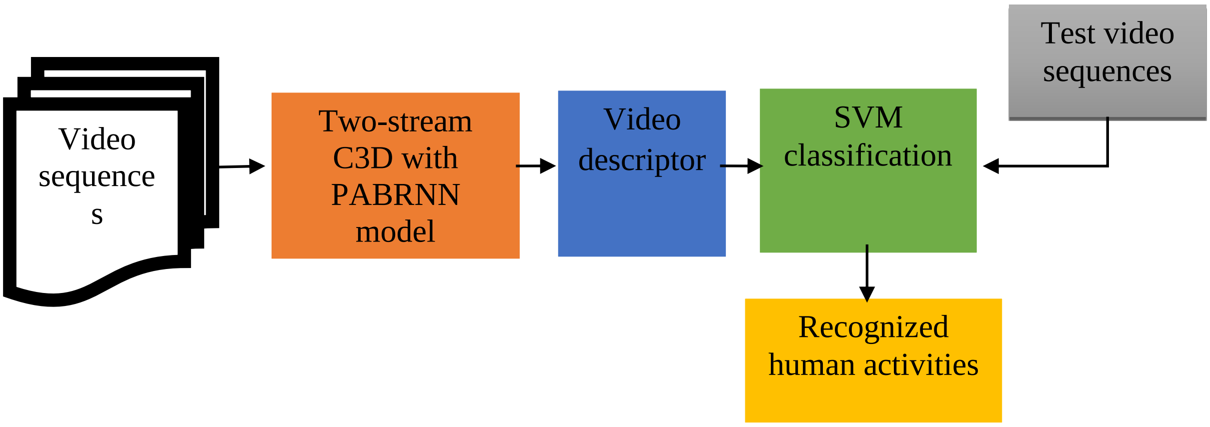

Figure 5.1. Block Diagram of JTDPABRD based HAR System

The two-stream C3D network begins by taking in data from a series of frames or clips from each video sequence. Both an attention stream and a feature stream are used as input in this network. Extraction of trajectory points (or optical flow) between clips is done in the feature stream, while extraction of joint guidance is done in the attention stream. Each channel's activations are combined with those of other channels to reveal the whole picture of the body's joints and trajectories. RNN, specifically the JTDRD approach, is used in place of max-min pooling to produce the pooled feature vectors associated with a single clip. However, the typical RNN has difficulty optimizing the aggregation of network outputs when multiple networks have been trained using the same data. Therefore, the issues with the traditional RNN are addressed by employing a Video clips are used to train a JTD-Bidirectional RNN-Descriptor (JTDBRD) that can describe events both in the past and the future. The concept is to split a regular RNN's state neurons into forward and backward sections. There is no connection between the outputs of advance states and the inputs of backward states. Using state, input information from both the past and future of the currently estimated frames, the objective function can be reduced in real time.

This BRNN can be trained using the same methods as a standard RNN, as it requires only a small number of specialized solutions at the beginning and end of training samples. Both the input at time $t = 1$ to the forward state and the input at time $t = T$ to the reverse state cannot be verified. However, they are randomized to a fixed value (0.5). Furthermore, as the information beyond the advance states' $t = T$ and the backward states' $t = 1$ is not relevant for the present update, the local state derivatives at those times have been set to 0. However, BRNN can't be used to provide the most likely important feature vectors. Another issue with BRNN is how to combine feature representations using hidden vectors. Therefore, PABRNN is proposed by this JTDPABRD method, with the idea being that if a feature from one video frame also occurs in another video frame, then the two together provide contextual aid. That is, nearby features are more likely to provide information about body joints and trajectories than distant ones. Figure 5.2 depicts the whole two-stream C3B framework that may be trained using PABRNN.

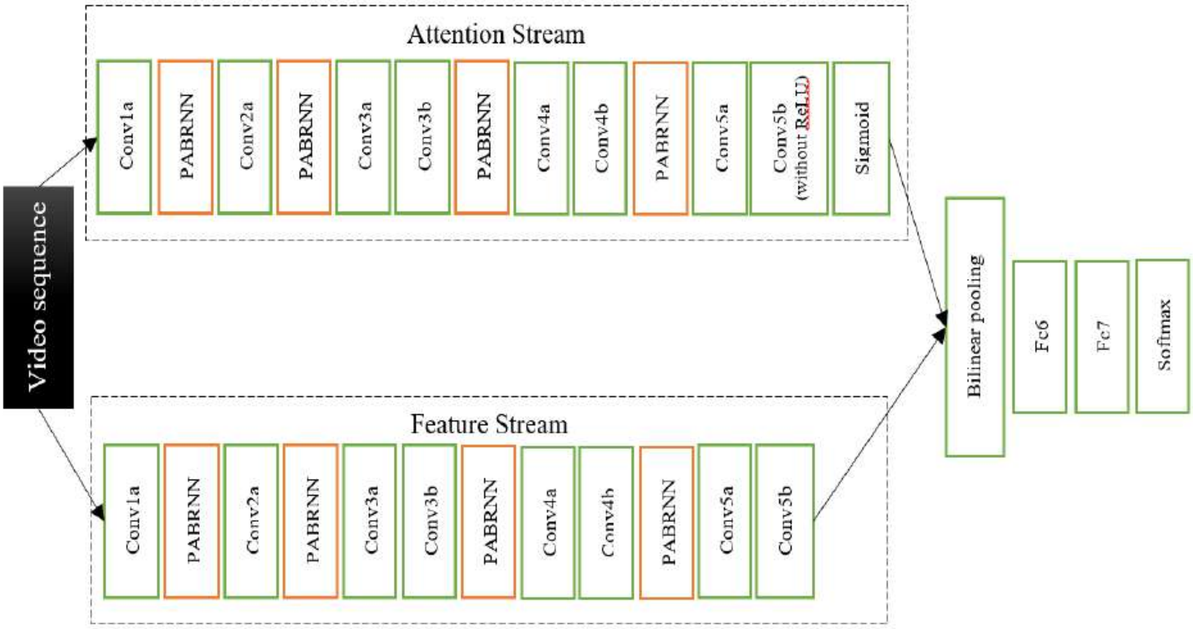

Figure 5.2. Block Diagram of Two-stream Bilinear C3D with PABRNN-based Feature Aggregation Method

5.1.1 PABRNN Model

In this PABRNN, the feature vector representation is a BRNN, which employs the pre-trained feature embeddings (body joints and trajectory points) as input to construct the hidden vectors via recurrent updates. The feature vector representations are aggregated using standard attention, and the attentive weights are generated in large part by the hidden vectors. To do this, a positional attention system was suggesting and recommend certain extra steps beyond the purview of typical attention.

- Locate the frames within a video series that contain a given occurrence feature.
- The feature vector guidance should be propagated in a position-aware manner.
- Create position-aware guidance vectors for all clip features based on the spread advice.
- Integrating the position-aware guidance vector into the already-established focus method.

Using the meticulous representations of both the original and aggregated feature vectors, the relevance between each dimension is assessed using a variety of similarity functions. As a similarity function, l_1-norm is often paired with the Manhattan distance (sim).

$$sim(F, F_a) = e^{-(\|F-F_a\|_1)} \tag{5.1}$$

The $\|\cdot\|_1$ is the l_1-norm, is used in Eq. (5.1), along with the original feature vector, F, and the aggregated feature vector, F_a, for each video clip.

Figure 5.3 depicts the organizational scheme of this PABRNN system.

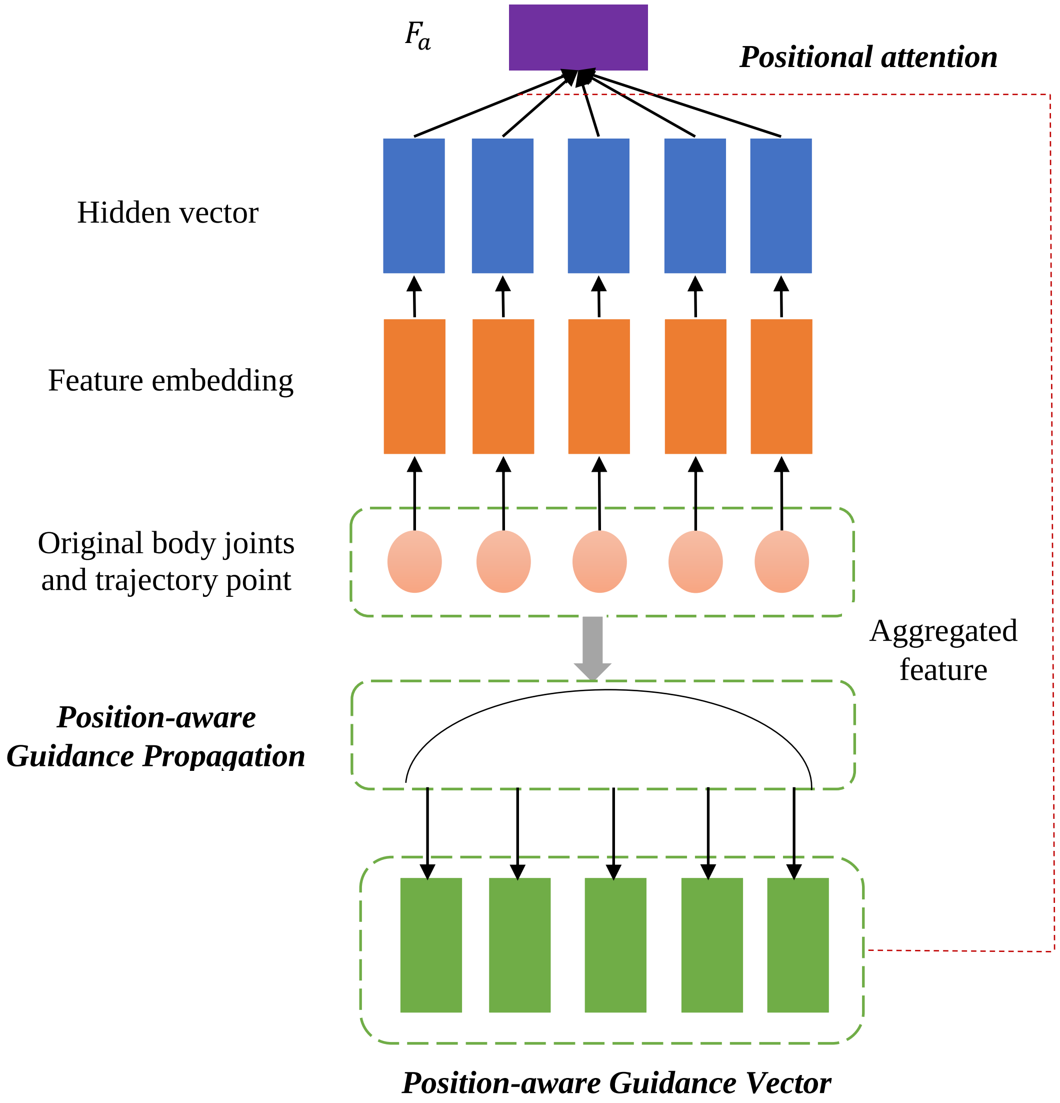
F_a
Positional attention
Hidden vector
Feature embedding
Original body joints and trajectory point
Aggregated feature
Position-aware Guidance Propagation
Position-aware Guidance Vector

Figure 5.3. Feature Vector Representation with PABRNN Framework

5.1.2 Position-aware Guidance Propagation

All of this is taken into account so that the features can be guided by the surrounding context if they show up in other clips. Here, the Gaussian kernel is used to simulate the spread of position-aware guidance, and the model reads as follows:

$$Kernel(d) = e^{\left(-d^2/_{2\sigma^2}\right)} \tag{5.2}$$

The kernel-based guidance acquired at a given distance d is denoted as $Kernel(d)$ in Eq. (5.2), where d is the distance between the original and aggregated features, σ is a parameter that constrains the propagation scope, and K is the kernel. As the distance increases, the position-aware steering appears to weaken. The highest level of propagating guidance is achieved in particular at $d = 0$. In this case, the feature vectors are all given the same value σ, and Integrating locational context into focus is a major area of study.

5.1.3 Position-aware Guidance Vector

By acquiring the position-aware guiding vector for each feature vector in the videos, it is able to describe the steering of attentions in a high-dimensional space. It is recommended that the guidance for a specific distance be evaluated in relation to the Gaussian distributions over the hidden dimensions. Based on this assumption, a guidance base matrix G is created, with each column containing the guidance base vector associated with a certain distance. The element of G are outlined as follows:

$$G(i,d) \sim N(Kernel(d), \sigma') \tag{5.3}$$

In Eq. (5.3), N is the normal density with a $Kernel(d)$ prediction and a standard deviation of σ', and $G(i,d)$ is the advice connected to the distance of d at thei^{th} position. The guidance base matrix is used to determine the guiding vector for a feature at a specific location by summing the guidance of all features present in the relevant video frames.

$$A_j = Gc_j \tag{5.4}$$

The number of features at various distances is estimated using the distance count vector c_j in Eq. (5.4), and the aggregated guidance vector for the feature at position j is denoted by A_j. If a feature is located at location j, then the number of body joint and trajectory point features within d units of j is denoted by $c_j(d)$.

$$c_j(d) = \sum_{f \in F}[(j - d) \in pos(f)] + [(j + d) \in pos(f)] \qquad (5.5)$$

In Eq. (5.5), F stands for the multi-feature 3D feature maps, f stands for a body joint location or a trajectory point feature in F, $pos(f)$ stands for the set of all the clips in which f occurs, and [·] stands for an indictor function with a value of 1 if the condition is met and 0 otherwise.

5.1.4 Positional Attention

To incorporate the features' position-aware advice into the attentive representations of the aggregated features, a positional attention method is presented. In particular, a feature's focus weight at the j^{th} position in the combined feature vector is formulated as follows.

$$\alpha_j = \frac{e^{\left(e\left(h_j, A_j\right)\right)}}{\sum_{k=1}^{l} e^{\left(e\left(h_k, A_k\right)\right)}} \qquad (5.6)$$

The l length of the video sequence, the BRNN-based hidden vector at position j, h_j, the aggregated position-aware guidance vector, A_j, obtained in Eq. (5.4), and the score function, $e(\cdot)$, which estimates the feature significance based on the hidden vector and the position-aware guidance vector, are all shown in Eq. (5.6). After that, the score function is written as:

$$e\left(h_j, A_j\right) = v^T tanh\left(W_H h_j + W_A A_j + b\right) \qquad (5.7)$$

The components of Eq. (5.7) are 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. Following this, the aggregated feature vector (the weighted sum of all the hidden vectors) is given the attention weights that were calculated.

$$F_a = \sum_{j=1}^{l} \alpha_j h_j \qquad (5.8)$$

Therefore, the clip descriptors are obtained by summing all of the feature vectors associated with a single clip.

Algorithm:

Input: Video sequences from Penn Action Dataset

Output: Extracted body points, trajectory points (Video descriptor)

```
Begin
Split video sequences into clips;
$\boldsymbol{for}(each\ clip)$
Initialize CNN parameters for both attention and feature streams;
Compute all the activations in convolutional layers;
Aggregate activations of each convolutional layers using PABRNN;
//PABRNN
Formulate the position-aware guidance propagation via Gaussian kernel;
Calculate the guidance base matrix related to certain distance;
Aggregate the guidance of all features in convolutional layers;
Obtain the aggregated guidance vector;
Determine the score function and the attentive weight of features in the aggregated guidance vector;
Find the resultant aggregated feature vector belonging to one clip i.e., obtain the clip descriptors;
Combine attention and feature streams using bilinear product function;
Apply fully connected and softmax layer;
Train the C3D using aggregated guidance feature vector;
Predict the video descriptors for a video sequence;
Perform SVM classifier;
//SVM
Initialize the video descriptor $(v)$ to classify;
Consider the training set$H = \{(v_1, y_1), \dots, (v_n, y_n)\}; //y$: class
Consider $k$ nearest neighbors;
Decide$y_r \in \{-1,1\}$;
```

```
Find $k$ sample and train the SVM;
Classify $v$ and get the result$y_r$;
Return$y_r$;
Recognize the individual activities in particular video sequence;
End
```

To further improve its representational power, bilinear production were use to mix clip descriptors from many convolutional layers. The whole network is trained end-to-end with softmax loss supervised by class label by adding the clip descriptors together, i.e. convolutional feature vectors with bilinear product. Human activity in a given video sequence can be identified once video descriptors are gathered and fed into a SVM.

5.2 EXPERIMENTAL RESULTS

The JTDPABRD method is implemented in MATLAB 2017b, and its efficacy is evaluated in terms of recognition accuracy in contrast to the JTDBRD, JTDRD, and JTDD. The experiment uses the 2326 video sequences from 15 different activity classifications found in the Penn Action dataset. The clips are culled from a wide variety of video sharing sites. Each clip averages between 50 and 100 still images in duration. There are 13 annotated body joints per frame.

To ensure accuracy, only 20% of the data is used for testing, while 80% is used for training. Baselines are the body's joint coordinates, the points along the trajectory, and the C3D characteristics. Therefore, JTDPABRD is tested using numerous configurations of pooling, or feature aggregation, to see which is most effective.

Measures of recognition accuracy include the True Positive (TP) and True Negative (TN) rates as a percentage of total trials. That's because:

$$Reg\_Acc = \frac{TP+TN}{TP+FP+FN+TN} \quad (5.9)$$

In Eq. (5.9), FP and FN stand for "false positive" and "false negative," respectively. Total Recognized Amount (TP) is the sum of all lawful actions that have been properly identified as such. The number of officially recognized illegal activities that are also illegal is denoted by TN. The FP is the total quantity of things that people think are lawful but are actually illegal. The FN amount of things that are thought to be prohibited but are actually legal. The results of both the body joint extraction and the trajectory point extraction are shown in Figure 5.5.

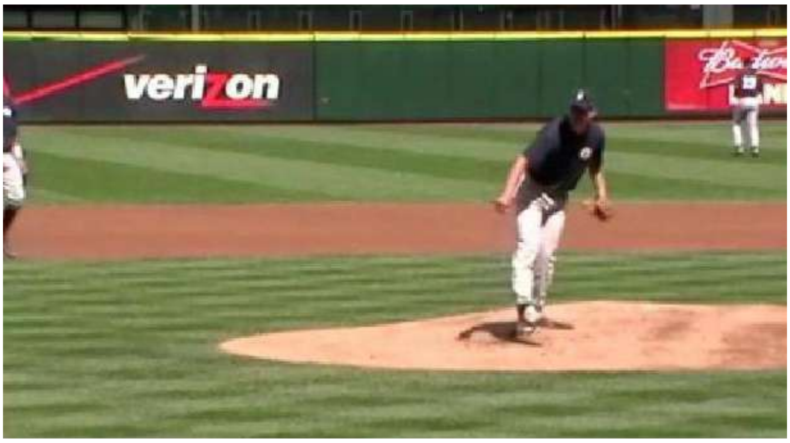

Figure 5.4 (a). Sample Input Video Sequence

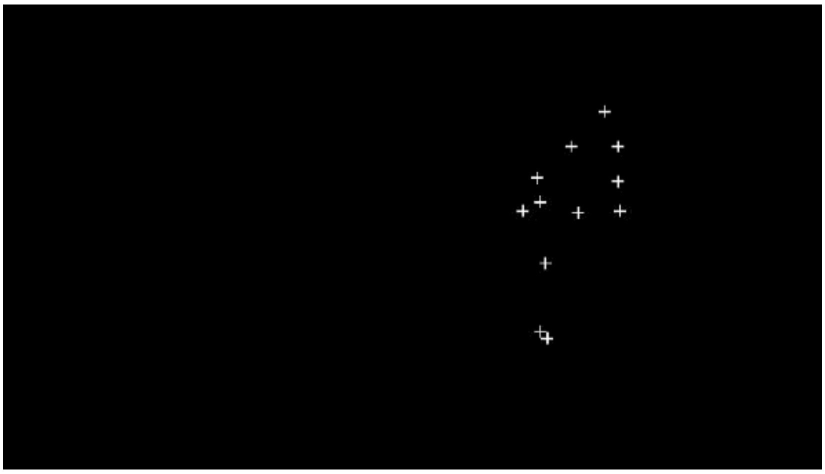

Figure 5.4 (b). Image for Body Joints Extraction from Input Video Sequence

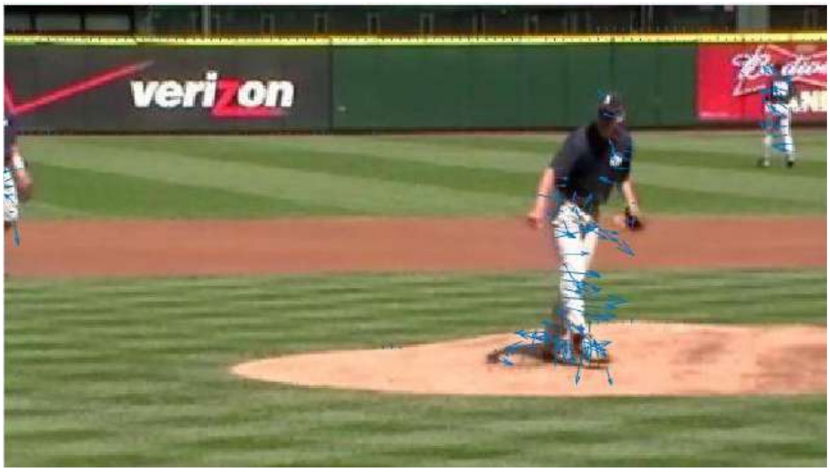

Figure 5.4 (c). Results for Trajectory Points Extraction of Input Video Sequence

Table 5.1 displays the results of the recognition accuracy analysis performed on the Penn Action dataset. Body joint coordinate recognition accuracies using trajectory point coordinates as a feature and C3D features are shown in the first column of Table 5.1. The results of these experiments demonstrate that using only the positions of body joints and the direction of optical flow as characteristics is insufficient. C3D features that combine all activations in a given layer are said to be particularly discriminative due to their high level of performance. $fc7$ has slightly less reliable recognition than $fc6$. Since the current implementation of C3D on the Penn Action dataset lacks the ability to fine-tune the $fc7$ layer, which is ideal for constructing the video descriptor on the pre-learned dataset, this is theoretically achievable. Many body joint and trajectory point features are aggregated using PABRNN across multiple 3D $conv$ layers to produce JTDPABRD.

Table 5.1. Recognition Accuracy of Baselines and JTDPABRD with Various Configurations on Penn Action Dataset

	Concatenate all the activations	**JTDPABRD Ratio Scaling (1×1×1)**	**JTDPABRD Coordinate Mapping (1×1×1)**	**JTDPABRD Ratio Scaling (3×3×3)**	**JTDPABRD Coordinate Mapping (3×3×3)**
Joint coordinates + trajectory coordinates	0.6452	-	-	-	-
$fc7$	0.7638	-	-	-	-
$fc6$	0.7811	-	-	-	-
$conv5b$	0.7345	0.8358	0.8829	0.8385	0.8683
$conv5a$	0.6675	0.7768	0.8047	0.7722	0.7831
$conv4b$	0.5684	0.7965	0.7873	0.8135	0.8258
$conv3b$	0.4602	0.7268	0.7059	0.7336	0.7315

As can be shown in Table 5.1, the JTDPABRD performs better than the JTDBRD, JTDRD, and JTDD when it comes to averaging the guided feature vectors of body joints and trajectory points in a video sequence. In addition, JTDPABRDs from different $conv$ layers are combined for processing to determine whether or not they can strike a fair balance. Combinations of convolution layers and fully connected layers are shown in Table 5.2 for the Penn Action dataset as fusion layer results.

Table 5.2. Activity Recognition Accuracy for Fusing Multiple Layers for Penn Action Dataset

Fusion Layers	JTDD	JTDRD	JTDBRD	JTDPABRD
	Recognition Accuracy			
$\boldsymbol{conv5b + fc6}$	0.867	0.871	0.875	0.883
$\boldsymbol{conv5b + conv4b}$	0.987	0.989	0.991	0.994
$\boldsymbol{conv5b + conv3b}$	0.873	0.875	0.879	0.883

Figure 5.5 shows how the feature extraction and identification results benefit greatly from the fusion of JTDPABRD over many layers. Combining $conv5b + conv4b$ into a JTDPABRD boosts individual activity recognition accuracy. This is due to the $conv$ layers aggregating more important features.

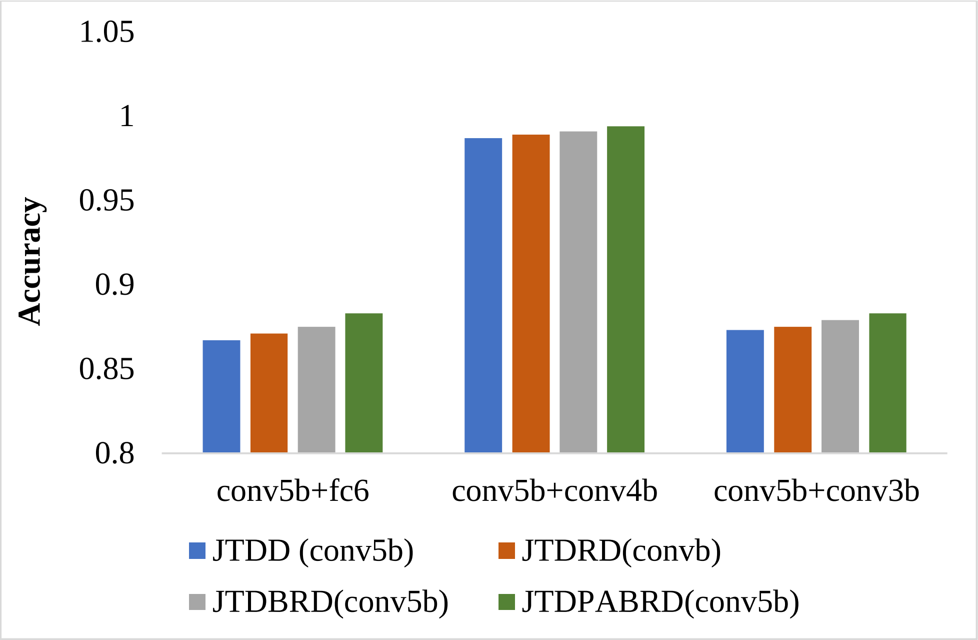

Figure 5.5. Recognition Accuracy of Fusing JTDPABRD from Multiple Layers together on Penn Action Dataset

The results of fusing many layers together on the Penn action dataset are shown in Table 5.3 in terms of precision, recall, and f-measure.

Table 5.3. Precision, Recall, and F-measure of Fusing Multiple Layers Together on Penn Action Dataset

Performance Metrics	Fusion Layers					
	$conv5b + fc6$		$conv5b + conv4b$		$conv5b + conv3b$	
	JTDD	**JTDPABRD**	**JTDD**	**JTDPABRD**	**JTDD**	**JTDPABRD**
Precision	0.855	0.874	0.975	0.983	0.856	0.871
Recall	0.861	0.880	0.982	0.991	0.869	0.878
F-measure	0.858	0.877	0.979	0.987	0.863	0.875

Figure 5.6 shows how the feature extraction and identification results benefit greatly from the fusion of JTDPABRD over many layers. Combining $conv5b + conv4b$ into a JTDPABRD boosts individual activity recognition of precision. This is due to the $conv$ layers aggregating more important features.

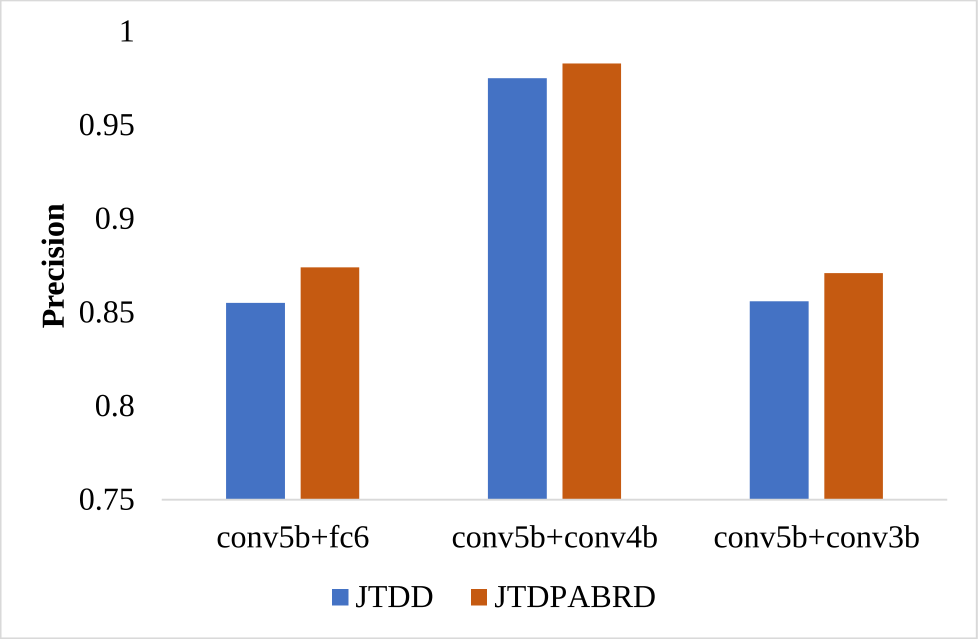

Figure 5.6. Recognition Precision of Fusing JTDPABRD from Multiple Layers together on Penn Action Dataset

Figure 5.7 shows how the feature extraction and identification results are significantly enhanced thanks to the fusion of JTDPABRD of different layers. Combining $conv5b + conv4b$ into a JTDPABRD boosts individual activity recognition of recall. This is due to the $conv$ layers aggregating more important features.

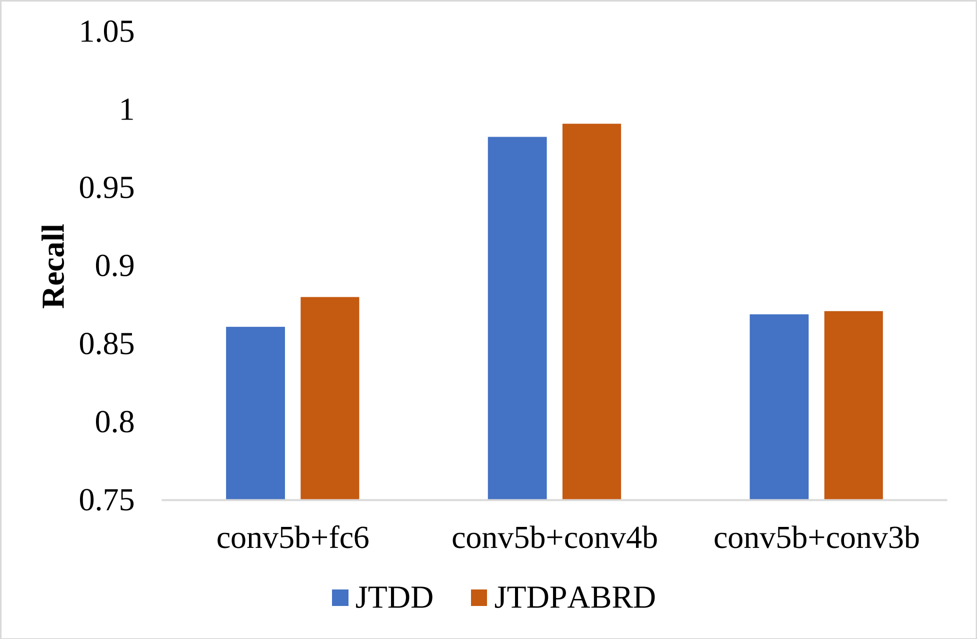

Figure 5.7. Recognition Recall of Fusing JTDPABRD from Multiple Layers together on Penn Action Dataset

Figure 5.8 shows how the feature extraction and identification results are significantly enhanced by the fusion of JTDPABRD of multiple layers. Combining $conv5b + conv4b$ into a JTDPABRD boosts individual activity recognition of F-measure. This is due to the $conv$ layers aggregating more important features.

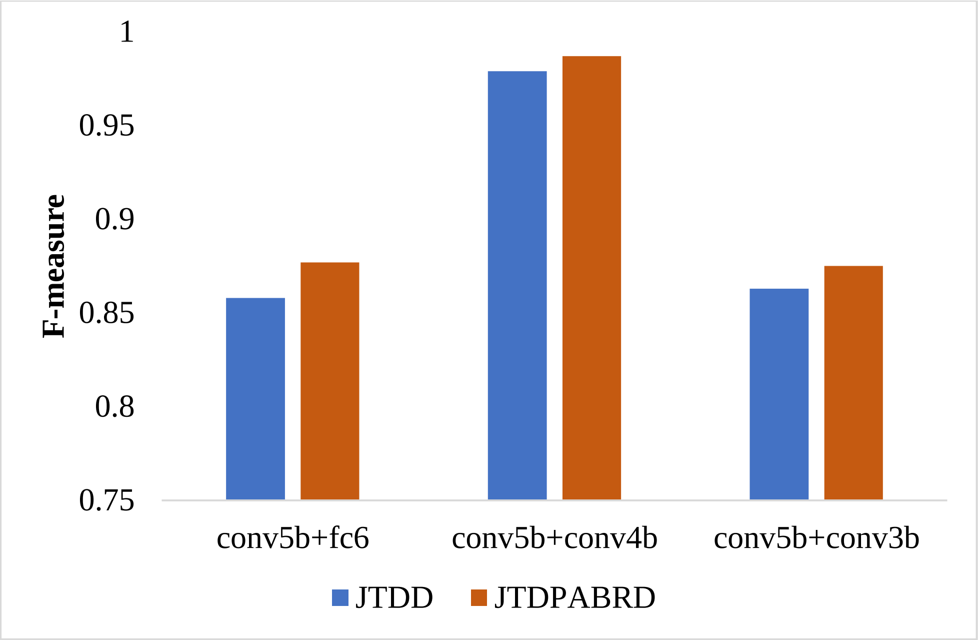

Figure 5.8. Recognition F-measure of Fusing JTDPABRD from Multiple Layers together on Penn Action Dataset

Multiple HAR methods' results on the Penn Action dataset are shown in Table 5.4 which draws parallels between predicted and ground-truth (GT) body joints and trajectory points.

Table 5.4. Impact of Estimated Body Joints + Trajectories versus GT Body Joints + Trajectories for Different Methods on Penn Action Dataset

Methods	GT	Estimated	Difference
JTDD $(\boldsymbol{conv5b})$	0.835	0.810	0.025
JTDRD $(\boldsymbol{conv5b})$	0.838	0.815	0.023
JTDBRD $(\boldsymbol{conv5b})$	0.843	0.821	0.022
JTDPABRD $(\boldsymbol{conv5b})$	0.847	0.828	0.019

Figure 5.9 demonstrates that the JTDPABRD outperforms the other methods on the Penn Action Dataset. The JTDPABRD results in the least variation between the GT body joints+ trajectory points and the estimated body joints+ trajectory points, in comparison to other methods.

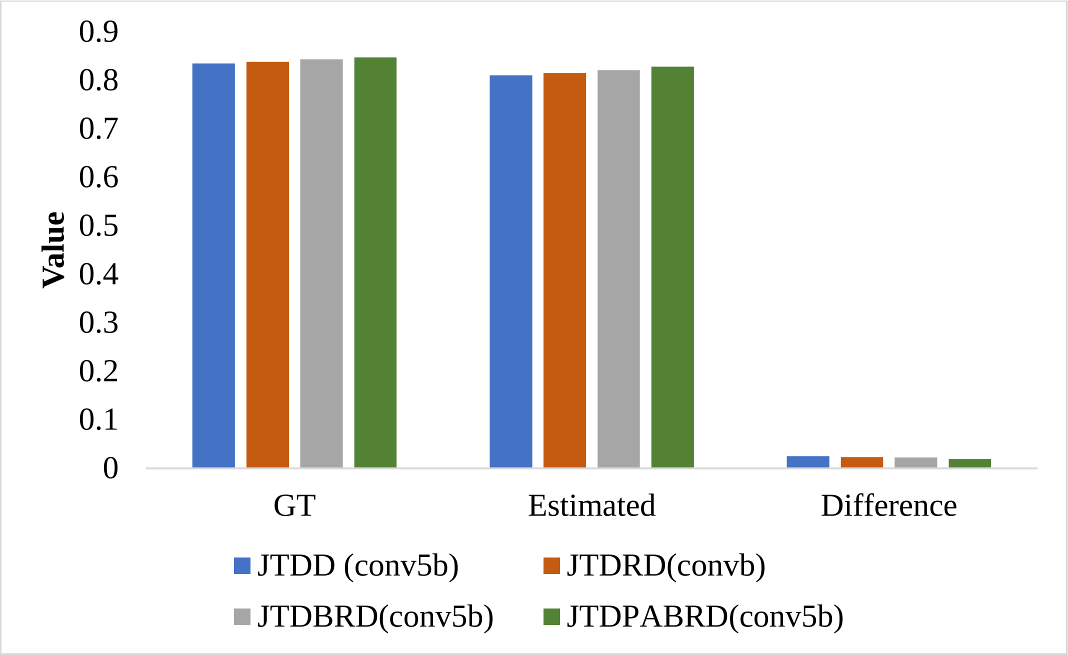

Figure 5.9 Impact of Estimated Body Joints + Trajectories versus GT Body Joints + Trajectories for Different Methods on Penn Action Dataset5.4 CHAPTER SUMMARY

5.3 CHAPTER SUMMARY

In this chapter, offer JTDPABRD, a combination of the PABRNN and the two-stream C3D network, for quickly extracting the necessary spatiotemporal information and boosting the accuracy of detecting individual actions. Multiple clips are extracted from the video before being fed into the two-stream C3D network. In a two-stream C3D network, the focus stream is responsible for extracting the body's joint locations, while the feature stream is responsible for extracting the points along the trajectory and any important spatiotemporal data. The convolutional feature vector representations of all clips in a single video are then aggregated to form the clip descriptor using the PABRNN. Furthermore, the entire pipeline, which consists of these two streams multiplied by a bilinear product, is trained using class labels. In addition, the activations of completely connected layers and the differences between them in space and time are combined to form the ultimate video description. To determine the action in a video, the SVM is provided this video description. The testing results showed that JTDPABRD approach utilizing Penn Action dataset achieves the highest recognition accuracy of 0.994 by fusing it from $conv5b$ and $conv4b$ with GT feature vectors.