International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Enhanced Tennis Video Analysis: R-CNN-Based Player Action Recognition and Event Detection with Dense Trajectories

Shama P. S.*[1], Prakash Pattan[2]

Submitted:13/03/2024 **Revised**: 28/04/2024 **Accepted**: 05/05/2024

Abstract: Recognizing tennis video actions, detecting balls and players, and tracking them pose challenges due to complex backgrounds, variable lighting, and camera movements. This study presents a highly sophisticated trajectory-based system for action recognition. The system, which integrates dense optical flow tracks, scale-invariant feature transform key points, the histogram of directed gradient, optical flow, and motion boundary histogram, is a testament to the complexity and depth of our research. The system includes ball detection, tracking, and event identification in tennis videos. The aim is to automatically annotate tennis matches, enabling low-cost visual sensing equipment to record and replay matches. The approach achieves an average overall accuracy of 84.34% in tennis video classification.

Keywords: *Tennis Video Analysis, Action Recognition, Optical Flow Tracking, Scale-Invariant Feature Transform, Histogram of Directed Gradients, Motion Boundary Histogram, Ball and Player Detection, Algorithmic Accuracy in Sports Analytics.*

1. Introduction

In recent decades, the availability of multimedia content has significantly expanded in our daily lives, and this trend is expected to continue. In response to this growth, content-based video analysis, indexing, and retrieval technology have become increasingly crucial. Sports videos have gained immense popularity, leading to substantial research in sports video content analysis. The proposed trajectory-based system for action recognition in tennis videos has the potential to revolutionize the field and inspire a new wave of research and innovation. Its applications, from automation and personalization to virtual advertisement insertion and 3-D virtual sports event generation, are vast and promising.

However, a semantic gap exists between the richness of user meanings and the simplicity of low-level visual and audio information in sports video analysis. An intermediate-level representation of sports video content is essential to bridge this gap. This is where sports video objects come into play. They serve as an effective mid-level representation that facilitates semantic analysis, enabling structure analysis, event identification, and tactic analysis across various sporting events, specifically focusing on tennis.

Tennis, a globally renowned sport, attracts millions of fans in attendance and through television broadcasts. Furthermore, statistical models have demonstrated their effectiveness in predicting match outcomes, emphasizing the significance of analyzing player actions during tennis matches. Recognizing players' activities is vital for match analysis, fostering technical and tactical coaching aid, and enhancing sports enthusiasts' understanding of the game.

One emerging area of interest in human action recognition is trajectory-based technology, which shows promise in capturing temporal correlations by tracking interest spots throughout a video. This technique involves extensive feature point sampling in each frame, followed by optical flow-based tracking. Multiple descriptors are computed along feature point trajectories to capture shape, appearance, and velocity information. Motion boundary histograms (MBH) have shown outstanding results due to their resistance to camera motion. While camera motion estimates can be beneficial in some cases, they can also generate irrelevant trajectories. Understanding camera motion speed can help prune trajectories, keeping only those relevant to people or objects of interest. Correcting optical flow based on camera speed also ensures that human motion vectors remain independent of camera movement, improving performance for motion descriptors based on optical flow.

This research has a secondary objective of exploring the benefits derived by coaches from implementing an event retrieval system to access automatically indexed events compared to traditional retrieval methods. Incorporating automatic event detection as part of instructional coaching sessions has revealed substantial improvements in participants' experiences. The system automatically monitors players and ball movements during tennis matches, detecting and tracking crucial events like tie-breaks. With abundant data, users can uncover intriguing play trends, including player and ball movement information. Coaches can query the data for critical match moments or identify player trends requiring attention.

[1]*Research Scholar, Dept. CSE, PDACE, Kalaburagi, Affiliated to Visvesvaraya Technological University, Belagavi, Karnataka, India https://orcid.org/0009-0000-4465-3407*
[2]*Assoc. Prof. Dept. of Computer Science & Engineering, PDACE, Kalaburagi, Karnataka, India*
** Corresponding Author Email: shamasg.publications@gmail.com*

In tennis, an event is defined by specific actions, such as a net approach, rally, ace, unreturned serve, or double fault. Rally and net approach events can be further categorized based on how players earn points, including passing the ball, moon ball, drop shot, unforced error, and volley. This study focuses on four key events, serve, bounce, hit, and net, to analyze their impact on the game.

Efficient and accurate ball tracking is crucial for building this system. Among various automated ball-tracking systems, this research employs advanced algorithms, particularly those suitable for camera-equipped quadcopters [1]. However, the challenges quadcopter-mounted cameras pose include air turbulence, rotor vibrations, and potential choppiness in the footage. Additional difficulties arise due to the small ball size (67mm diameter), high-speed movements (up to 225kph), varying illumination, multiple moving objects in frames, and similar object characteristics. To overcome these challenges, this paper proposes a system that integrates computer vision and machine learning approaches to develop a new algorithm that surpasses existing ones regarding accuracy and speed in ball tracking.

2. Literature Survey

Mohak Sukhwani et al. [2] developed a system for automatically generating frame-level fine-grained annotations, showcasing its application in tennis videos. This system employs probabilistic labeling, consistent sparse coding, dictionary learning, and the K-SVD algorithm to create detailed annotations using textual descriptions.

Alessandro Micarelli et al. [3] proposed an automatic annotation system for tennis video sequences, employing three key modules: frame selection, color-based filtering, and edge detection. This efficient method uses the Hough Transform to identify court lines and player positions, showcasing its potential for data scientists.

Fei Yan et al. [4] introduced a fully automatic annotation system for tennis matches using broadcast video. They enhanced their tennis ball tracking system with acoustic signal processing techniques and treated event categorization as a sequence labeling problem, evaluating various machine learning algorithms for performance.

Qingwu Li et al. [5] presented an improved video representation method using salient dense trajectories and motion boundary descriptors. This approach surpassed state-of-the-art results on standard video action datasets and was developed in collaboration with the University of Central Florida.

Tianyi Liu et al. [6] developed a video representation approach based on dense trajectories and motion boundary descriptors, emphasizing advanced optical flow techniques and the superiority of motion boundary histograms (MBH) for real-world videos with significant camera motion.

3. Methodology

We have introduced a novel tennis video action recognition approach based on dense trajectories for future applications, as shown in Figure 1. The methodology involves a frame extraction module that leverages k-means clustering and color-based segmentation to extract information from previous work. Hough line transformation is employed to detect the court region and identify court lines. The architectural framework consists of three primary modules, as depicted in Figure 2: (1) Feature extraction from dense trajectories, (2) ball detection and tracking, and (3) event detection using the R-CNN Classifier.

(a) (b)

Fig 1. (a) Video frame (b) Extracted dense trajectories (red points are interest points, green curves are trajectories)

In the dense trajectory module, feature points are sampled from each frame across various spatial scales, forming dense trajectories by tracking these points through median filtering and information derived from a dense optical flow field [7]. These trajectories are then utilized to compute various feature descriptors. HOG (histogram of oriented gradients) and MBH are employed among these descriptors. Subsequently, ball detection and tracking are accomplished through Homograph transformation. Finally, event classification is performed using the R-CNN deep learning classifier, categorizing four distinct events [8].

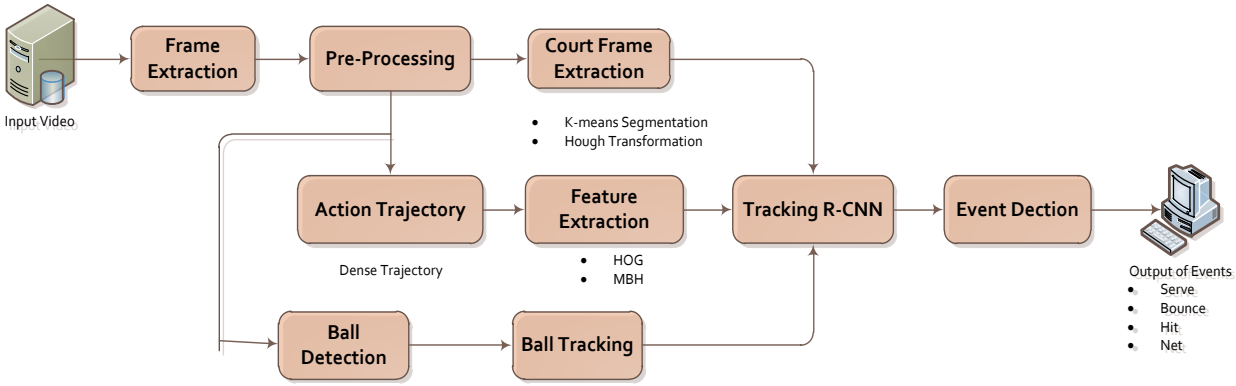

Fig 2. The architecture of the Proposed System

3.1. Dense trajectory

The dense trajectory technique extracts local descriptors along a moving human's path [9]. The key objectives of this technique are as follows:

1. Densely Sample Feature Points: A densely sampled set of feature points is computed in each video frame, ensuring comprehensive coverage.

2. Optical Flow Tracking: Optical flow tracks specific points within the video, allowing for monitoring feature points over time.

3. Essential Trajectories: Trajectories are fundamental in this approach, as they represent the paths followed by feature points across multiple frames.

4. Comprehensive Information: Multiple descriptors are computed along the trajectories of feature points, capturing information about shape, appearance, and motion.

These aspects collectively contribute to the dense trajectory technique's effectiveness in recognizing human activity and solving related problems in video analysis [10].

3.2. Dense sampling

In the Dense Trajectory approach, feature points are categorized into multiple groups based on the scale of the input image [11]. This multi-scale sampling ensures that feature points are adequately distributed across various spatial locations and scales. In most cases, experimental results have shown that eight spatial scales are sufficient, and feature extraction is performed separately on each scale [12].

A sampling interval, denoted as W = 5, is defined to identify the feature points' grid size. Subsequently, a region with an undefined structure, called a homogeneous region, is identified and removed. This step is crucial as feature points cannot be effectively tracked within these regions; hence, they are excluded from tracking. Points in homogeneous areas are removed following the criteria outlined by Shi and Tomasi, where points are eliminated if the eigenvalues of the auto-correlation matrix exhibit smaller values.

$$T = 0.001 \times max_{i \in I} \min(\lambda_i^1, \lambda_i^2) \quad (1)$$

Where $(\lambda_i^1, \lambda_i^2)$ represents eigenvalues i of point in the image I

3.3. Trajectories

The dense optical flow field is computed so that the sampled points. The main advantage of finding dense optical flow is that it permits fast and robust tracking and irregular motion patterns. For every frame of input video I_t Finding its dense optical flow element is $w_t = (u_t, v_t)$ concerning the next frame I_{t+1} 'Re' the optical flow components are represented as u_t and v_t Are horizontal and vertical components? Consider a position $Pt = (x_t, y_t)$ in frame I_t, its tracked position in the frame I_{t+1} is smoothed by applying a median filter onw_t:

$$P_{t+1} = (x_{t+1}, y_{t+1}) = (x_t, y_t) + \frac{(M * w_t)}{x_t, y_t} \quad (2)$$

Where M is the median filtering kernel, the median filter kernel M size is 3×3 pixels.

Points from all the successive frames are added to obtain the trajectories:$(P_t, P_{t+1}, P_{t+2} \ldots\ldots)$. We need to adjust the trajectory length to L = 15 frames because the dense sampling points can change their position in tracking. While tracking, if no new points are found along with $W \times W$ In the neighborhood, a new sampling point is added; therefore, dense coverage can be achieved.

3.4. Trajectory shape descriptor

To obtain the patterns, the shape of the trajectory is encoded as a local motion pattern. Taking the length L value, the sequence of shapes can be represented as $(\Delta P_t, \ldots\ldots, \Delta P_{t+L-1})$ With the displacement of:

$$\Delta P_t = (P_{t+1} - P_t) = (x_{t+1} - x_t, y_{t+1} - y_t) \quad (3)$$

The output vector can be rewritten as:

$$T = \frac{(\Delta P_t, \ldots\ldots, \Delta P_{t+L-1})}{\sum_{j=t}^{t+L-1} \left\| \Delta P_j \right\|} \quad (4)$$

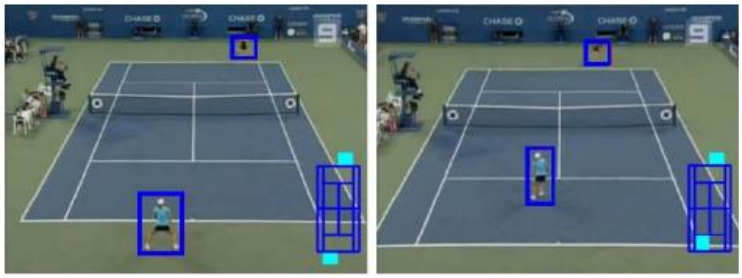

Fig 3. Player Detection

Along with extracting shape features using dense trajectory, utilize the dense optical flow obtained during the computation of dense sampling to recognize human action. Histogram of Oriented Gradient and Motion Boundary Histogram features are utilized. With the feature point, L, as the trajectory length, the NXN region surrounding the feature selected point is considered on every frame to form a time-space structure. With the structure of time-space and grid division, the entire region is divided into n_σ in all the directions, n_γ As a homogeneous time. Therefore, a total of $n_\sigma \times n_\sigma \times n_\gamma$ The region is utilized to extract the features in the selected time-space structure. Figure 3 shows the result of player detection.

3.5. Motion and Structure Descriptors

The HOG feature, which stands for Histogram of Oriented Gradients, is a method for calculating the gradient of a gray-scale image using histograms. The HOG feature is computed based on a histogram designed to have eight bins. Due to this choice of histogram bins, the length of the HOG feature is 96 (2 * 2 * 3 * 8). Figure 4 illustrates the flowchart depicting the process of computing the HOG feature.

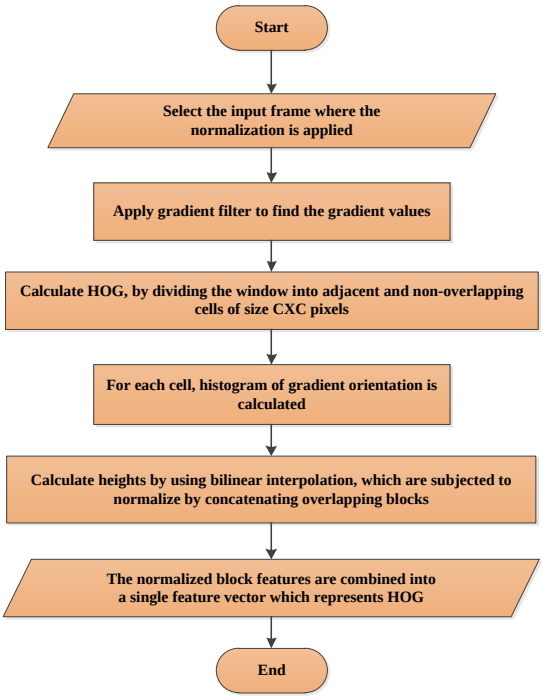

Fig 4. Computation HOG feature

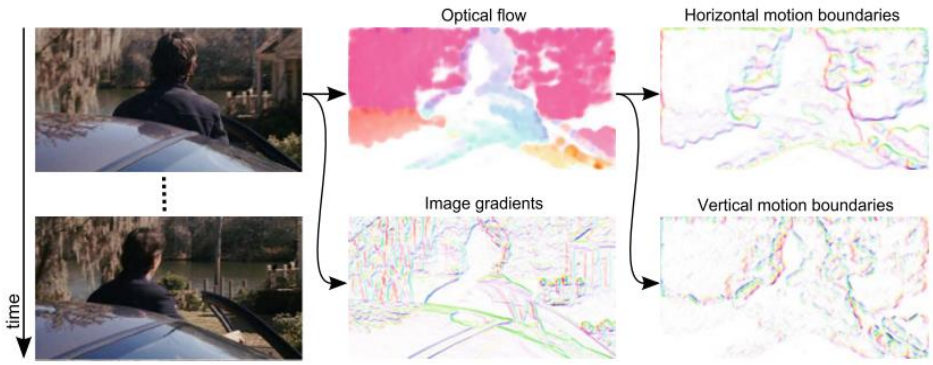

Fig 5. Illustration of the information captured by HOG and MBH descriptors.

Algorithm of Dense Trajectory
Inputs: Video frame **Output:** Trajectory Descriptors
Step.1 Sampling points with intersecting points of information are initially selected to calculate dense optical flow fields for an input video stream, and dense optical flow fields are then calculated.

Step.2 *In each frame, the foreground region, or region of interest, is determined by calculating the value of optical flow. The ROI is identified by selecting the points closely associated with the intersecting points. The discovered sample points unrelated to the ROI are removed, resulting in the determination of the trajectory's starting point.*

Step.3 *Using the foreground trajectory, points are tracked in eight scale spaces, by which the foreground trajectory is structured with a scaling factor between each pair of scales set to* $1/\sqrt{2}$. *In the proposed method, the trajectory length is initiated to 15 to overcome the problem of drifting of feature points.*

Step.4 *This procedure outputs a 3x3x2 block of space-time structure constructed from the foreground trajectory, from which a feature descriptor is extracted—the proposed method used by HOG and MBH.*

The MBH, or Motion Boundary Histogram, feature descriptor is computed for human recognition by independently calculating the derivatives of the optical flow's horizontal and vertical components. This descriptor contains information about the relative movement between pixels and uses two separate scalar maps for the horizontal and vertical motion components to distinguish between them [13]. In this process, the HOG is calculated for two optical flow elements, and differences in motion boundaries are retained. In contrast, data with constant motion is removed due to their minimal differences. This helps reduce or eliminate the effects of camera motion. After computing spatial derivatives in all directions, the results are represented as a histogram, with magnitude used for weighting. The MBH feature generates a pair of horizontal and vertical feature descriptors, resulting in a total length of 192 (2 * 96) features. Figure 5 represents the HOG and HOF features' gradient and optical flow information, respectively.

3.6. Ball Detection and Tracking

The ball-tracking operations of the algorithms are detailed as follows:

Corner Critical Point Matching: The algorithms utilize the Hamming distance to identify corner critical points within each frame. These vital points are then matched with corresponding essential points of adjacent frames. This matching process helps estimate the homography between crucial points. Homography transformation involves encoding eight parameters encompassing translation, rotation, scaling, skew, and perspective transformations occurring at a specific point. This is illustrated in the following example:

$$x' = Hx \tag{6}$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \tag{7}$$

Where x defines the critical point of frame A, x' is a crucial point of the second frame, and B and H represent the value of homography. The matrix format of H can be written as:

$$H = \begin{bmatrix} s.a\cos\theta & s.b\sin\theta & t_x \\ s.a\sin\theta & s.b\cos\theta & t_y \\ P_x & P_y & 1 \end{bmatrix} \tag{8}$$

Where s defines the scaling factor, a and bRe skew parameters, rotation factors are $\sin\theta$ and$\cos\theta$, t_x and t_yAre translation elements, P_x and P_yBe the perspective transformation.

The process of identifying moving foreground objects in the video involves the following steps:

1. Temporal Differencing: Moving foreground objects are detected by analyzing the temporal differences between adjacent frames. Subsequently, a morphological opening operation is applied to identify the objects in motion.

2. Object Classification: Each identified moving foreground object is categorized as either a ball or a non-ball object based on specific characteristics such as size, shape, and gradient direction within the object's boundary.

3. Sliding Window Technique: A temporary sliding window is employed, with centers at each frame "i," and the windows are rotated from frame "i-V" to frame "i+V." At each point within the frame, a small ellipsoid circle is defined in the column-row-time-space, encompassing one point from frame "i-1" and one from frame "i+1." If such locations exist, the three points within the circle are termed a "seed triplet" and are fitted using a constant acceleration dynamic model.

This process is repeated iteratively and greedily until convergence is achieved, ensuring that all object points are identified. Convergence is determined based on the cost parameter "λ," which should no longer decrease. "λ" is defined as follows:

$$\lambda = \sum_{j=i-V}^{i+V} \sum_{k} \rho(P_j^k) \tag{9}$$

Along with each object point cost:

$$\rho\left(P_j^k\right) = \begin{cases} d^2\left(P_j^{\wedge\square}, P_j^k\right) & if\ d\left(P_j^{\wedge\square}, P_j^k\right) < d_{th} \\ {d_{th}}^2 & if\ \ d\left(P_j^{\wedge\square}, P_j^k\right) \le d_{th} \end{cases} \quad (10)$$

Where P_j^k is the observed position of the kth ball candidate in frame j, $P_j^{\wedge\square}$ In the current model, d gives the estimated ball position in frame j (.,.), the Euclidean distance, and d^{th} is a predefined threshold.

In the process, a cluster of candidate pixels is generated with each movement and rotation of a sliding window. These clusters may be linked to a candidate object or from background clutter. Weighted and directed graphs are then constructed, where each node within the graph represents a candidate position. The distance between two nodes is determined based on the compatibility between the candidate object pixels associated with each node. As illustrated in Figure 6, the ball's trajectories are computed by identifying all pairs of nodes that pertain to the candidate pixels.

Fig 6. Ball Tracking

3.7. Event Detection

In a tennis match, significant events such as the tennis ball being struck, bouncing on the court, or hitting the net mark essential moments in the game. Observing discontinuities in the tennis ball's motion makes it possible to detect these events. An event is identified and flagged when orientation and motion magnitude changes exceed predefined thresholds along the trajectory. Once this process is completed, a machine-learning model uses the feature vector generated for each window as input for event classification.

The proposed CNN model comprises eight layers of varying sizes and types, organized as follows (from the top layer to the bottom layer) as shown in Table 1:

Table 1. Confusion Matrix of Event Detection

Layer Type	Parameters	Description
Top Layer	-	Initial input layer
Second Layer	-	Additional input processing
Third Layer	-	Further input processing
Fourth Layer	-	Pre-convolutional processing
Convolution	5 X 5 X 48	Convolutional layer with 48 filters, 5x5 kernel
MaxPool	2 X 2	Max pooling layer with 2x2 pool size
Convolution	5 X 5 X 48	Convolutional layer with 48 filters, 5x5 kernel
MaxPool	2 X 2	Max pooling layer with 2x2 pool size
Convolution	5 X 5 X 24	Convolutional layer with 24 filters, 5x5 kernel
Dense	64 neurons	A fully connected layer with 64 neurons
Softmax	3 classes	Output layer with softmax activation for 7 classes

The final dense layer utilizes a sigmoid activation function, while the output layer provides probabilities for four distinct labels, as detailed below. Convolutional and other dense layers use the rectified linear unit (ReLU) as the activation function. The architecture of this CNN was developed through a trial-and-error process, and the paper delves into L2 regularization and stochastic gradient descent for CNN training. The classifier is trained to recognize and classify tennis events such as serving, bouncing, hitting, and striking the net.

3.8. Regions with CNN features (R-CNN)

Deep models are employed for object detection by utilizing region-based convolutional neural networks (R-CNNs), also referred to as regions with CNN features (R-CNNs). This

innovative approach to object detection applies deep models to the problem at hand. R-CNN models select multiple proposed areas from an image and assign categories and bounding boxes to these regions. CNN is deployed to perform forward computations on the data to extract features from each selected area of interest. Subsequently, the features extracted from each proposed region are utilized to make predictions regarding their categories and bounding boxes. The algorithm for object classification is elaborated upon in the following section. Figure 7 shows the flowchart of RCNN [14].

Algorithm of Region-based Convolutional neural networks

Inputs: Feature Set

Output: object classification

Step.1 *The input image is subjected to a selective search algorithm to select multiple regions for further analysis. These regions are chosen based on various scales and shapes and size and shape variations. Different regions of interest are labeled based on the type of data and the ground truth.*

Step.2 *A trained CNN is used in a truncated form and placed before the output layer to improve performance. An ROI is converted into input dimensions required by the network, and features extracted from the ROI are output using forward computation performed on the ROI data.*

Step.3 *A concatenation of the feature extracted and labeled types for each ROI is used to train multiple support vector machines for object classification. Specifically, each support vector machine determines whether or not an example falls into a specific category.*

Step.4 *To predict ground-truth bounding boxes, each region of interest's features and labeled bounding box are concatenated and fed into a linear regression model trained on the data.*

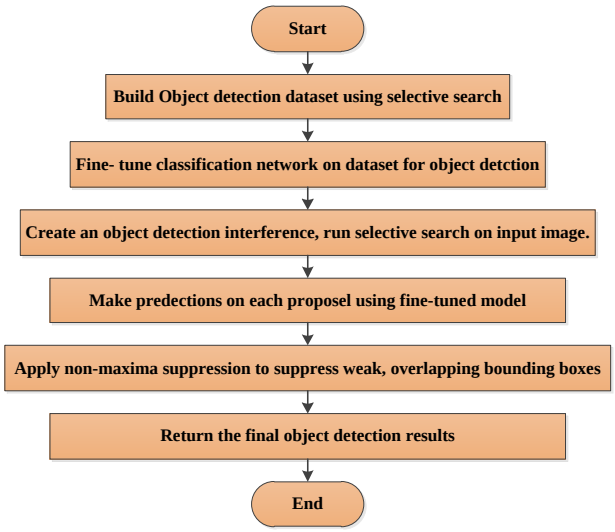

Fig 7. Flowchart of R-CNN

4. Experimental Results

Numerous experiments were conducted, as exemplified in Figure 8, to assess the efficacy of the proposed methods using a video sequence from a tennis tournament. The video sequences featured in the top row encompass a total of 25 videos contained within the database. Each video spans one minute, exhibiting a frame rate of 30 frames per second and a resolution of 640 x 480. Notably, every video comprises approximately 1800 frames, signifying a substantial volume of data. These videos encompass a blend of synthetic and natural settings. Figure 9, conversely, provides a visual representation of the ball's position and trajectory throughout the game's timeline, corresponding to the frame number. Within this illustration, the ball's present location is denoted by the yellow circle, while the green line traces the ball's trajectory.

Frame #30 Frame #60 Frame #90

Frame #80 Frame #110 Frame #250

Fig 8. Examples of Video frames

Frame #30 Frame #320 Frame #793

Fig 9. Ball tracking Trajectory

Figure 10 illustrates the current position and trajectory of the players relative to the current frame number. In this depiction:

- The blue circle represents the position of the players.

- The yellow line signifies the trajectory of the lower half player.

- Meanwhile, the green line tracks the trajectory of the upper half player.

Frame #26 Frame #503 Frame #912

Fig 10. Player tracking Trajectory

5. Performance Measure

Plotting the ground truth values enables us to assess the proposed model's performance in event identification. This system can detect all four events within a play and furnishes users with comprehensive evaluation results. Figure 11 presents the performance metrics regarding precision, recall, and F-measure [15]. Furthermore, Table 2 provides an overview of the effectiveness of various event detection types.

Table 2. Confusion Matrix of Event Detection

	Bounce	**Hit**	**Net**	**Total**
Bounce	29	3	3	35
Hit	3	35	4	42
Net	2	3	33	38

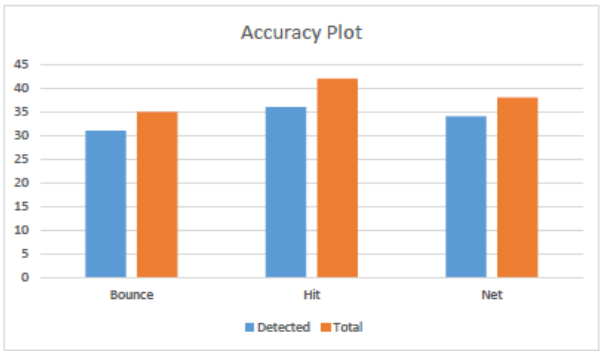

Fig 11. Effectiveness of Different Types of Event Detection

$$Accuracy\ (in\ \%) = \left(\frac{Accurately\ Recognized\ Events}{Total\ Number\ of\ Events} \right) \times 100 \qquad (11)$$

Equation (11) provides the calculation for system accuracy as a percentage.

Accuracy Bounce Detection: (29/35) * 100% = 82.85%

Accuracy Hit Detection: (35/42) * 100% = 83.33%

Accuracy Net Detection: (33/38) * 100% = 86.84%

Overall, Event Detection Accuracy is 84.34%.

Table 2. State of Art Comparison

Algorithms	Accuracy (%)
SVM	79.88%
CNN	81.21%
R-CNN	84.34%

Table 2 presents a comparative analysis of the state-of-the-art algorithms in terms of their accuracy. The table includes three widely recognized algorithms: Support Vector Machine (SVM), Convolutional Neural Network (CNN), and Region-based Convolutional Neural Network (R-CNN). In this comparison, SVM shows an accuracy of 79.88%, which is commendable but lower than the other two. CNN, known for its proficiency in handling image data, demonstrates a slightly higher accuracy of 81.21%. However, the most accurate of the three is R-CNN, an advanced version of CNN, which shows a significant improvement with an accuracy of 84.34%. Table 2 clearly illustrates the advancements in algorithmic accuracy, highlighting R-CNN's superior performance in this specific context.

6. Conclusions

Sports video analysis tools are gaining popularity, enabling more precise visualization and game action analysis. This paper successfully demonstrates all four types of event detection. It discusses the unique approaches developed for player action identification, ball detection, and tracking. Furthermore, the proposed model incorporates several improvements to handle challenges such as monitoring small tennis balls and variations in camera motion. These enhancements enhance ball recognition accuracy and calculate the ball's anticipated trajectory location.

Conflicts of interest

The authors of this manuscript state that they have no financial, personal, or professional conflicts of interest that could have influenced the work reported in this paper. This declaration encompasses all forms of potential disputes, ensuring the integrity and impartiality of the research and its findings.

References

[1] Tayeba Qazi, Prerana Mukherjee, Siddharth Srivastava, Brejesh Lall, Nathi Ram Chauhan, Department of Mechanical and Automation Engineering, Indira Gandhi Delhi Technical University for Women Delhi, India, "Automated Ball Tracking in Tennis Videos," 2015 Third International Conference on Image Information Processing. DOI: 10.1109/ICIIP.2015.7414796

[2] Mohak Sukhwani, C.V. Jawahar, "Frame Level Annotations for Tennis Videos," International Conference on Pattern Recognition (ICPR), Cancún Center, Cancún, México, 2016. DOI: 10.1109/ICPR.2016.7899932

[3] Alessandro Micarelli, Enver Sangineto, "Automatic Annotation of Tennis Video Sequences," Pattern Recognition, 24th DAGM Symposium, Zurich, Switzerland, September 16-18, 2002, Proceedings. DOI: 10.1007/3-540-45732-1_26

[4] Fei Yan, Josef Kittler, David Windridge, William Christmas, Krystian Mikolajczyk, Stephen Cox, Qiang Huang, "Automatic annotation of tennis games: An integration of audio, vision, and learning," Elsevier Image and Vision Computing, 32 (2014) 896–903. DOI: 10.1016/j.imavis.2014.01.004

[5] Qingwu Li, Haisu Cheng, Yan Zhou, and Guanying Huo, "Human Action Recognition Using Improved Salient Dense Trajectories," Hindawi Publishing Corporation Computational Intelligence and Neuroscience, Volume 2016, Article ID 6750459, 11 pages. DOI: 10.1155/2016/6750459

[6] Tianyi Liu, Shuangsang Fang, Yuehui Zhao, Peng Wang, Jun Zhang, "Implementation of Training Convolutional Neural Networks," University of Chinese Academy of Sciences, Beijing, China.

[7] Heng Wang, Alexander Kläser, Cordelia Schmid, Cheng-Lin Liu, "Dense Trajectories and Motion Boundary Descriptors for Action Recognition," International Journal of Computer Vision (2013) 103:60–79. DOI: 10.1007/s11263-012-0594-8

[8] Ehtesham Hassan, Yasser Khalilm, and Imtiaz Ahmad, "Learning Feature Fusion in Deep Learning-Based Object Detector," Hindawi, Journal of Engineering, Volume 2020, Article ID 7286187, 11 pages. DOI: 10.1155/2020/7286187

[9] Liqin Huang, Xiangyu Zhang, and Wei Li, "Dense Trajectories and DHOG for Classification of Viewpoints from Echocardiogram Videos," Hindawi Publishing Corporation Computational and

Mathematical Methods in Medicine, Volume 2016, Article ID 9610192, 7 pages. DOI: 10.1155/2016/9610192

[10] Srikanth Muralidharan, Mehrsan Javan, Greg Mori, "Learning features from Improved Dense Trajectories using deep convolutional networks for Human Activity Recognition."

[11] Heng Wang, Alexander Kläser, Cordelia Schmid, Cheng-Lin Liu, "Action Recognition by Dense Trajectories."

[12] Manjunath Jogin, Divya G D, Mohana, Meghana R K, Madhulika M S, Apoorva S, "Feature Extraction using Convolution Neural Networks (CNN) and Deep Learning," 2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT-2018), MAY 18th & 19th 2018. DOI: 10.1109/RTEICT.2018.9012507

[13] Do Hang Nga, Yoshiyuki Kawano, Keiji Yanai, "Fusion of Dense SURF Triangulation Features and Dense Trajectory-based Features."

[14] Gowda, S., Murthy, S., Hiremath, J., Belur Subramanya, S., S. Hiremath, S., & S. Hiremath, M. (2024). Activity recognition based on spatio-temporal features with transfer learning. *IAES International Journal of Artificial Intelligence (IJ-AI), 13*(2), 2102-2110. doi:http://doi.org/10.11591/ijai.v13.i2.pp2102-2110

[15] M. S. Hiremath, R. C. Biradar, L. Subhadarshini, S. S. Hiremath, J. S. Hiremath and S. K. Sivanandan, "Enhancing Alzheimer's Disease Diagnosis: A Hybrid Model of Transfer Learning and SVM for Improved Classification Accuracy," 2024 IEEE 9th International Conference for Convergence in Technology (I2CT), Pune, India, 2024, pp. 1-8, doi: 10.1109/I2CT61223.2024.10543710.

Bibliography

Mrs. Shama P.S. earned her B.E. degree in Information Science Engineering from P.D.A College of Engineering, Gulbarga, Karnataka, and her M.Tech in Computer Science & Engineering from VTU PG Centre, Gulbarga, Karnataka. She is a Research Scholar in the Department of Computer Science & Engineering at P.D.A College of Engineering, Gulbarga, Karnataka. With over seven years of experience in academia and research, her research focuses on Image Processing and Deep Learning techniques. She has contributed to 5 journal publications and has one conference publication to her credit.

A Professor in the CSE Department, he brings over 30 years of rich experience encompassing teaching and research. He is adept at coordinating research projects and seminars. He holds an M.Tech in Information Technology from Jawaharlal Nehru Technological University, Hyderabad, India, and earned his Ph.D. in 2015. His research expertise spans Digital Image Processing, Computer Vision, the Internet of Things, and Computer Networks. He has published over 30 research papers in esteemed journals and holds a patent. His contributions include organizing technical events and serving as a reviewer for international journals. Notably, he has guided several research scholars to their Ph.D. degrees and mentored two more in computer vision.