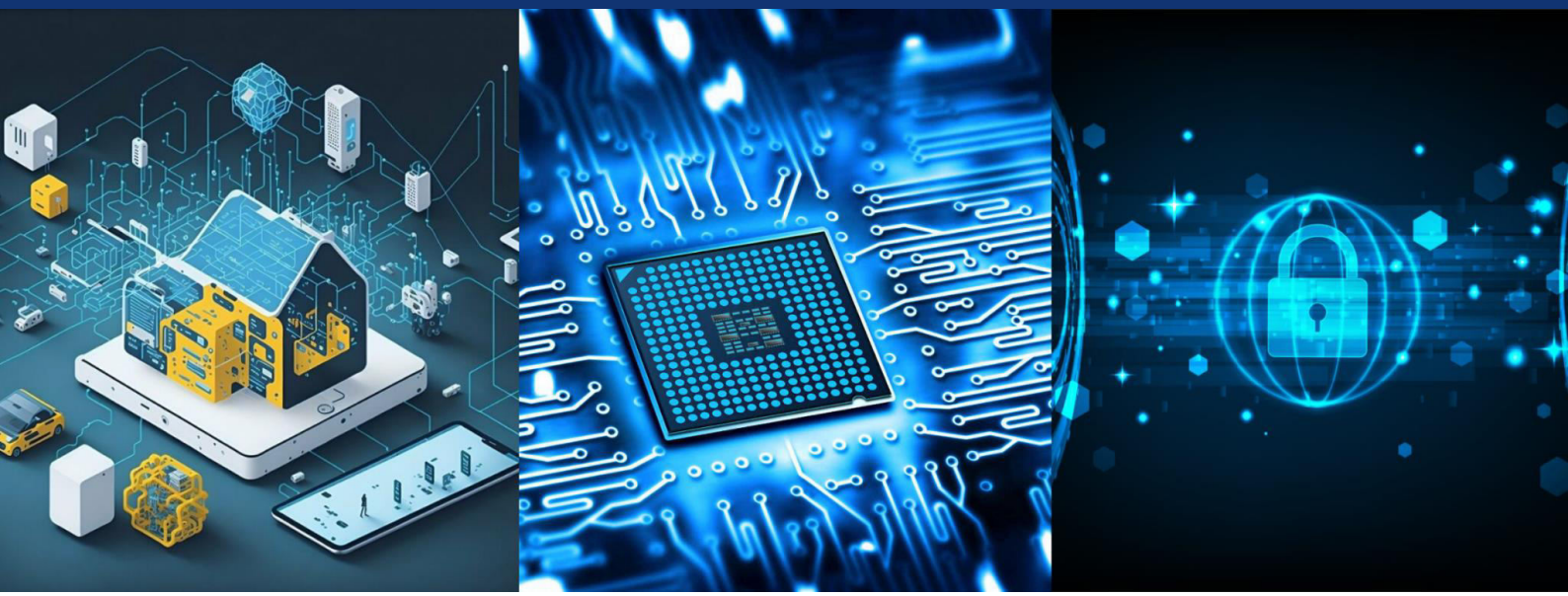
IJIRCCE

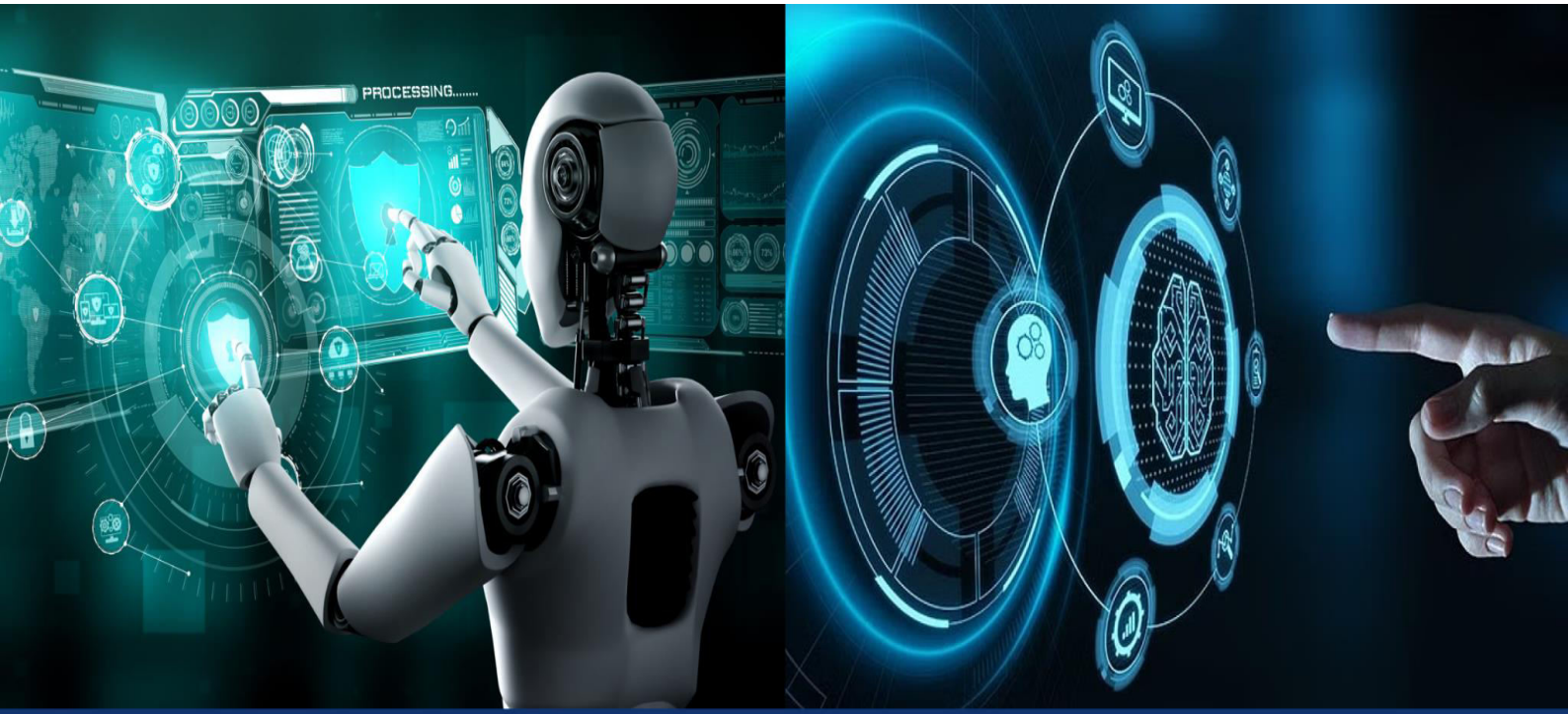
ISSN(O): 2320-9801

ISSN(P): 2320-9798



International Journal of Innovative Research in Computer and Communication Engineering

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.771

Volume 13, Issue 4, April 2025

© 2025 IJIRCCE | Volume 13, Issue 4, April 2025 |DOI:10.15680/IJIRCCE.2025.1304247

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Tennis Analysis System

Shruti Gotsurve, Aditya Patil, Prathmesh Patil, Vaishnavi Patil, Prof.S.Chunamari

Department of Artificial Intelligence and Data Science, A.C.Patil College of Engineering, Kharghar, Navi Mumbai, India

ABSTRACT: This paper presents a real-time Tennis Analysis System leveraging the YOLO (You Only Look Once) object detection algorithm. The system is designed to analyze tennis matches by accurately detecting and tracking key objects such as the tennis ball, players, and racket movements. Utilizing YOLO's fast and efficient detection capabilities, the system captures critical events like serve speeds, shot directions, and player positioning with high precision. The model is trained on a specialized dataset tailored for tennis, ensuring robust performance in diverse environments such as indoor and outdoor courts. This framework offers insights into player performance, match statistics, and tactical analysis, providing valuable data for coaches, analysts, and players. The real-time analysis feature makes it suitable for live broadcasts, enhancing the viewer experience through enriched game commentary and statistics. Future extensions include integrating pose estimation for improved action recognition and expanding the system's applicability to other racket sports

I.INTRODUCTION

In modern sports science, accurately monitoring and analyzing the action states of athletes is crucial for enhancing training outcomes and competitive performance. This is particularly significant in tennis, where the player's actions, such as standing, moving, and striking, directly affect the outcome of the match. Traditional methods of action analysis, which rely on manual observation and recording, are time-consuming and susceptible to subjective biases. Thus, developing an efficient and accurate automated action recognition system has become a key direction in sports science research. template..In recent years, deep learning technology has made significant advances in the field of computer vision, especially with object detection algorithms like YOLO (You Only Look Once), which have opened new possibilities for real-time action recognition. YOLO achieves rapid and precise object detection by predicting the location and category of targets in a single pass. However, relying solely on YOLO for action recognition presents certain limitations when dealing with complex movements and subtle differences. This Is Another Level 2 Heading

Early research in the field of action recognition exhibited several shortcomings, including limited diversity and low annotation quality of datasets, an over-reliance on handcrafted features with high computational complexity, and a lack of real-time applicability and suitability for real-world scenarios. Additionally, these studies often struggled to accurately handle complex movements and athlete interactions, and they lacked interdisciplinary approaches that integrate knowledge from related fields. Furthermore, many of these studies did not undergo large-scale validation and overlooked the needs of actual users, which constrained their effectiveness and generalizability in practical applications.

To date, deep learning methods, particularly those involving computer vision techniques such as YOLO and RestNet, have been widely applied to various sports for tasks like player tracking, action recognition, and performance analysis. These methods have proven effective in sports like soccer, basketball, and athletics, where they are used to monitor player movements, analyze strategies, and even assist in real-time decision-making during games. By automatically detecting and classifying actions, these techniques have significantly improved the accuracy and efficiency of sports analysis, providing valuable insights for coaches and analysts. Despite the success of these methods in other sports, they have not yet been extensively applied to the analysis of tennis matches.

The unique dynamics of tennis, including the fast-paced exchanges, frequent player movements, and the involvement of both the player and the ball, present distinct challenges that have yet to be fully addressed by current deep learning techniques. This gap indicates a significant opportunity for research and development in applying these advanced methods to tennis, potentially leading to new tools and insights that could revolutionize how the sport is analyzed and understood

© 2025 IJIRCCE | Volume 13, Issue 4, April 2025 |DOI:10.15680/IJIRCCE.2025.1304247

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

II. RELATED WORK

YOLO (You Only Look Once) is a popular real-time object detection system developed by Joseph Redmon and others[1]. Its core idea is to use a single neural network to predict object bounding boxes and class probabilities, achieving fast and accurate object detection in images. YOLO is characterized by its speed and ability to process video streams in real-time, making it highly suitable for applications that require immediate response, such as surveillance and autonomous driving.

Regarding the research of YOLO in the field of sports science and game monitoring, several papers have demonstrated its application results in motion recognition and motion tracking. For example, one study used YOLO and Deep SORT technology to improve accuracy in soccer multi-detection tasks, achieving 95 percent ball detection accuracy through a semi-supervised learning system, which is significantly better than previous methods [2].

In addition, for basketball games, the researchers improved the ability to track and detect basketball games by integrating multi-source motion features and hybrid YOLO-T2LSTM networks [3]. Moreover, a study in the sport of squash evaluated multiple open sources, pre-trained deep convolutional neural networks suitable for detecting athlete movements from single-camera video to help coaches and players optimize training and competition strategies. These studies show that YOLO can not only track athletes and objects in real-time but also support the assessment of tactical placement and physical status by analyzing the dynamic position and movement patterns of athletes

If we want to achieve situational awareness of the entire tennis game, player behavior detection is quite an important problem that should be solved. In most cases, we conclude that players have three behaviors, which are moving, standing, and hitting. We can do some work base on the player's behavior. Two methods came out.

The first is to detect the player's motion straightforward base on computer vision, and we concluded that the training dataset is a segment of the video. The second method is detecting the behavior base player's pose, which training datasets are the details of the pose, including relative angle, absolute position vector, etc.

Obviously, we cannot just display the player's status straightly on screen, so Yolov5 has been introduced, first bracket the player, and write the state near the bracket. Mediapipe is a cross-platform open-source framework widely used for building pipelines for multimodal data, such as visual and audio data. For tennis player action recognition, Mediapipe can be used to capture the skeletal key points of the players (i.e., pose estimation). The pose estimation module of Mediapipe can detect the key points of the player in each frame of the video, including joints, limb positions, and more. These key points are then extracted to form a time-series data sequence, serving as input for further processing

The object detection plays a major part in determining the location of the ball in the particular video sequence. There are many popular deep learning based object detection methods available that can efficiently detect the ball and output a bounding box [1]. Some algorithms will demand a lot of computation power and some will be giving lesser accuracy. Before starting with detecting the ball trajectory has to be detected and for that various detecting algorithms are compared.

A. Single Shot Detection (SSD) Single Shot Detection is a method of object detection to analyse multiple objects by taking a single image. Various objects in an image are detected and analysed from a single frame. This is a much quicker method of analysis compared to Convoluted Neural Networks [3]. A feature layer of m*n is achieved for analysis for p channels. A bounding box is created for k regions. SSD is mostly called as a Multibox detector as we will compute each bounding box and we will compute offsets that are relative to the original bounding boxes. In a single bounding box different layers of feature maps are also created. The technique that is used is called as bounding box regression technique. The SSD architecture is built on a venerable VGG-16 architecture. This is a simple architecture that uses simple 3*3 convoluted networks that are stacked over each other to increase the depth of the image thereby reducing the volume which is handled by max pooling.

Multi box loss = confidence loss + alpha * location loss

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

B. You Look Only Once(YOLO) This is a commonly used algorithm that uses a single Neural Network for analysing the object in an image. This algorithm uses a single network for both classification and localisation of objects in an image. This algorithm also uses bounding boxes. There will be two bounding boxes for every grid in an image. This algorithm is much quicker compared to other algorithms. This understands generalized object representation that allows it to train the network on real time images and detect the objects quite accurately [1]. The speed varies from around 45 to 155 frames per second. The algorithm is unlike the region convoluted neural networks that perform the detection on different regions and thereby as a result the algorithm ends up predicting an image multiple times for different regions in an image. Unlikely, in the YOLO algorithm it is a Fully Convoluted Neural Network that passes the image only once and the output is predicted as the next step. The images that are trained are the full image and this optimizes the performance of the algorithm for the detection of objects in an image. The information of the image that is trained in the entire image and the details of the images are analysed globally. This is the most commonly used algorithm that is used for detecting natural images as this is the most efficient algorithm for new domains and inputs that are unexpected. Each boundary box is predicted from details of the entire image.

C. You Look Only Once(YOLO) This is a commonly used algorithm that uses a single Neural Network for analysing the object in an image. This algorithm uses a single network for both classification and localisation of objects in an image. This algorithm also uses bounding boxes. There will be two bounding boxes for every grid in an image. This algorithm is much quicker compared to other algorithms. This understands generalized object representation that allows it to train the network on real time images and detect the objects quite accurately [1]. The speed varies from around 45 to 155 frames per second. The algorithm is unlike the region convoluted neural networks that perform the detection on different regions and thereby as a result the algorithm ends up predicting an image multiple times for different regions in an image. Unlikely, in the YOLO algorithm it is a Fully Convoluted Neural Network that passes the image only once and the output is predicted as the next step. The images that are trained are the full image and this optimizes the performance of the algorithm for the detection of objects in an image. The information of the image that is trained in the entire image and the details of the images are analysed globally. This is the most commonly used algorithm that is used for detecting natural images as this is the most efficient algorithm for new domains and inputs that are unexpected. Each boundary box is predicted from details of the entire image.

D. Faster - Regional Convoluted Neural Networks (Faster RCNN) This algorithm is used to detect objects in an image by using two network regions- a region and a proposal network (RPN) and a detector network. The region network generates region proposals where the object is to be analyzed and the network region uses these proposals to detect the objects in an image [2]. A faster RCNN network creates generate regional proposals using selective search where the time cost for generating the regional proposals is much smaller compared to selective approach. The Regional Proportional Network (RPN) is the network that is responsible for the detection of the objects in an image. The 978-1-7281-3250-1/19/$31.00 ©2019 IEEE major role in a faster RCNN is played by the anchors which are boxes. A Faster RCNN has 9 anchors at different images at a given position [2]. The network uses various boxes where the boxes are analyzed by a classifier and the occurrence of objects is checked by the regressor. The CNN network detects an object class and the bounding box. The feature map for the picture is achieved after the images go through many convoluted areas. The location of each image is then made to pass through by a sliding window where all the features are extracted by the network for detection. The main work of the RPN network is to check whether the location contains an object and then the bounding boxes will pass to the detector for further detection and as a result it returns the object of the bounding box.

III. METHODOLOGY

A. Data collection

The data is collected in a tennis court which there are many cameras that are placed at different angles to record the images and videos. The camera is placed on a tripod that is at the height of an average human. The camera is placed at a height where the toss of a tennis ball can be easily detected and the angles of the camera can track the ball from the toss of the ball to the ball moving to the next court. Distribution of data being an important part for any deep learning based models; we collected data in such a way that there exist variations in every instance of data. There were data at various angles and lighting conditions and we collected data from 5 test subjects. Each videos are converted into frames and the frames are individually processed. Aspect ratio is an important thing that is needed to kept in mind before undergoing any preprocessing methods. If aspect ratio is changes by various degrees of resizing and forced dimensional reductions it will result in the loss o originality of the particular image and will let to training in false direction. To overcome this, the image is cropped individually into a square image and then set to processing. The colour and

brightness is maintained for generalization of the model. Bounding boxes are marked on the image and the results are saved in an encoded document. As the ball being circular, bounding boxes are mostly square and the ball images with

and the frames are individually processed. Aspect ratio is an important thing that is needed to kept in mind before undergoing any preprocessing methods. If aspect ratio is changes by various degrees of resizing and forced dimensional reductions it will result in the loss o originality of the particular image and will let to training in false direction. To overcome this, the image is cropped individually into a square image and then set to processing. The colour and brightness is maintained for generalization of the model. Bounding boxes are marked on the image and the results are saved in an encoded document. As the ball being circular, bounding boxes are mostly square and the ball images with reduced visibility is removed to increase the precision of the model. monotonicity is maintained in order to train each model with the same set of images.

The data is collected in a tennis court which there are many cameras that are placed at different angles to record the images and videos. The camera is placed on a tripod that is at the height of an average human. The camera is placed at a height where the toss of a tennis ball can be easily detected and the angles of the camera can track the ball from the toss of the ball to the ball moving to the next court. Distribution of data being an important part for any deep learning based models; we collected data in such a way that there exist variations in every instance of data. There were data at various angles and lighting conditions and we collected data from 5 test subjects. Each videos are converted into frames and the frames are individually processed. Aspect ratio is an important thing that is needed to kept in mind before undergoing any preprocessing methods. If aspect ratio is changes by various degrees of resizing and forced dimensional reductions it will result in the loss o originality of the particular image and will let to training in false direction. To overcome this, the image is cropped individually into a square image and then set to processing. The colour and brightness is maintained for generalization of the model. Bounding boxes are marked on the image and the results are saved in an encoded document. As the ball being circular, bounding boxes are mostly square and the ball images with reduced visibility is removed to increase the precision of the model. monotonicity is maintained in order to train each model with the same set of images.

The combination of YOLO and skeletal recognition is typically achieved through the following steps: First, the YOLO model is used to detect and locate athletes in the video, generating bounding boxes to separate the athletes from the background. Then, the image regions containing the athletes are cropped and fed into a skeletal recognition model, such as Mediapipe or a ResNet-based pose estimation model, to identify the key joints of the athletes. Finally, the results of skeletal recognition are combined with YOLO's detection outputs to achieve more accurate action classification, such as standing, moving, or hitting. This approach effectively combines YOLO's [1] object detection capabilities with skeletal recognition for action analysis, providing a powerful tool for automating the recognition and analysis of complex movements. These time-series data (sequential graph structures) are input into the ST-GCN model, where ST-GCN uses spatiotemporal convolution operations to extract spatial and temporal features. The extracted spatiotemporal features are then fed into a classifier, typically a fully connected network, to perform action classification. Ultimately, the system can identify specific actions being performed by the tennis player, such as swinging or running These time-series data (sequential graph structures) are input into the ST-GCN model, where STGCN uses spatiotemporal convolution operations to extract spatial and temporal features. The extracted spatiotemporal features are then fed into a classifier, typically a fully connected network, to perform action classification. Ultimately, the system can identify specific actions being performed by the tennis player, such as swinging or running

B. Data Preparing

For Resnet50, First, take any tennis match video data online before cutting off non-related parts and only save the downside of each data. Make sure the video is 30 frames per second. Next, make a category of video (1 and 0 represent hitting or other, respectively), convert it from video to picture by using cv2, and group it into 2 different files. Finally, use Yolo to extract the player into two 1(hitting) and 0(other) directories. To overcome oversampling, take one sample frame by 10 frames. For 3dResenet50, it is mostly the same, but the training dataset should be 5-second video; use a video that is 60 frames per second that can help 3dResent50 "connect" each frame more accurately while the processing speeds may reduced. For the third method, by using the STGCN model, it should first use Mediapipe to convert labeled data into skeleton parameter files (.npy). The skeleton parameter includes the relative angle between arms and body and the joint vector of shoulder, hand, leg, and barycenter. However, it is important to collect at least 1080p resolution of video data to train the opponent player, or the 3dResnet50 model may be divergent, as the camera angle is fixed, and the opponent player occupies a small number of pixels. Before sending data into different models, data enhancement is needed, especially on opponent training. Data enhancement can improve the universality of a model; also, because of the lack of training data, data enhancement can increase the amount of training data. Meanwhile, without data enhancement, the model cannot well recognize the opponent player's status.

© 2025 IJIRCCE | Volume 13, Issue 4, April 2025 |DOI:10.15680/IJIRCCE.2025.1304247

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

C. TRAINING

All the three models are trained with the processed data in a high powered Nvidia Geforce Gtx 1080 Ti Fe with 16GB of RAM. The learning rate is maintained the same for all the models for comparison and the learning rate is reduced after 1000 steps. Loss function off FRCNN reduced below 2 after 20,000 steps and for SSD below 50,000 steps and for YOLO below 40000. The models have settled but the training is continued till 2,00,000 steps. The models even after settling are trained up to 2,00,000 steps to achieve much stability and efficiency. There were around 5000 images that were given as an input to the model and the model was validated with approximately 100 images.

The whole periods start with after the data enhancement. We are aiming to make the name of the directory become the label 1 hitting and 0 other, but there are some uncertainties in single frames, as when cutting off some of the frames, human reaction time should be calculated. So, remove the front and back frames before putting them into the model. We applied a data enhancement method for Resent50, which is resize to height 224 and width 224. For three color channels, we normalize it as a mean equal to 0.485, 0.456, and 0.406, with standard deviations of 0.229, 0.224, and 0.225, respectively.

IV. ARCHITECTURE OF SYSTEM

A. TrackNet

TrackNet Architecture for Tennis Analysis,TrackNet is a deep learning model designed for tracking tennis balls in real-time. It uses an encoder-decoder structure with a fully convolutional network (FCN) to pro- cess three consecutive video frames and predict a heatmap showing the ball's position in the middle frame. This architecture leverages temporal informa- tionto handle fast movements, blurs, and occlusions. The encoder extracts spatial features, and the decoder generates pixel-wise heatmaps for precise localization. The MSE loss helps train the model to accurately predict the ball's location, supporting tasks like shot classification, speed tracking, and rally analysis

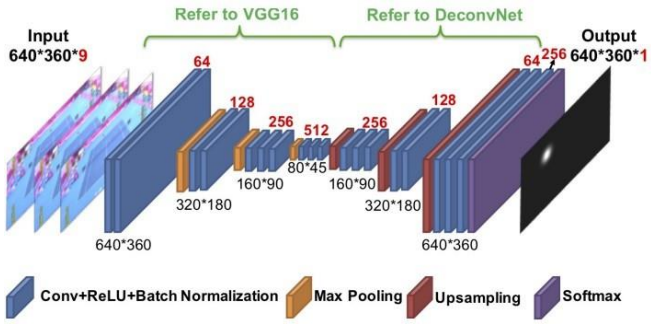


Figure : TrackNet Architecture

B. ByteTrack

ByteTrack is a robust multi-object tracking (MOT) algorithm that focuses on associating both high- and low-confidence detections across video frames to achieve improved tracking performance. Unlike conventional track-ers, which typically discard low-confidence detections, ByteTrack leverages every detection box to maintain better object tracking even in challenging conditions such as occlusions, fast movements, or crowd scenarios.

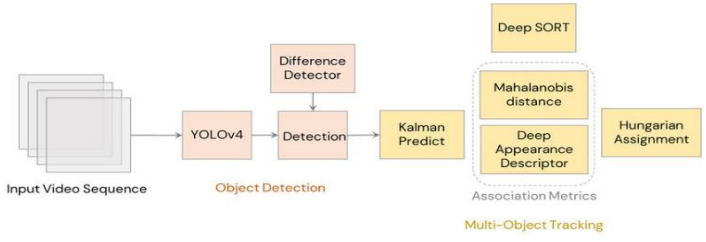


Figure: Byte-Track: Multi-Object Tracking by Associating Every Detection Box

© 2025 IJIRCCE | Volume 13, Issue 4, April 2025 |DOI:10.15680/IJIRCCE.2025.1304247

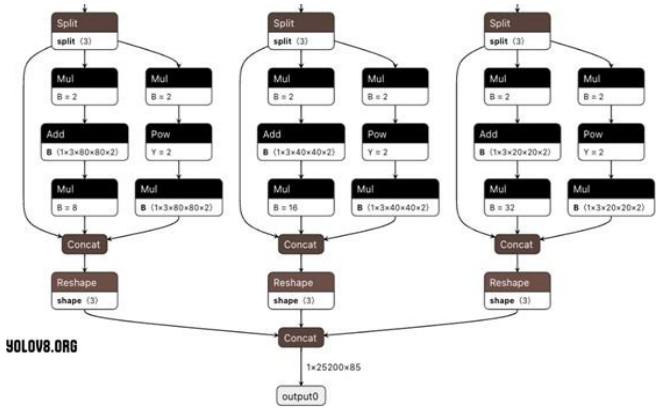
www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

C. YOLOV8

YOLOv8 is the latest version [23-24] of the YOLO (You Only Look Once)models. The YOLO models are popular for their accuracy and compact size. It is astate-of-the-art model that could be trained on any powerful or low-end hardware.Alternatively, they can also be trained and deployed on the cloud. The first YOLO model was introduced in a C repository called Darknet in 2015 by JosephRedmond [15] when he was working on it as PHD at the University of Washington. It has since been developed by the community for subsequent versions



V.HARDWARE AND SOFTWARE REQUIREMENTS

A. Hardware Requirements

Computing Resources : A high-performance laptop or desktop with at least 8GB of RAM and a multi-core CPU Preferably, a system with a dedicated GPU (e.g., NVIDIA) for deep learning model training

Storage : At least 100GB of available storage space for dataset storage and processing

B. Software Requirements

Programming Language : Python, Javascript, HTML, CSS

Libraries and Frameworks : YOLO Framework: Darknet, TensorFlow, or Py- Torch. OpenCV: For video processing and object tracking. NumPy/Pandas: For data manipulation and analysis. Matplotlib/Seaborn: For visualization of analysis results.

Development Environment : Jupyter Notebook or any Python IDE (e.g., Py- Charm, VS Code)

Version Control : Git and GitHub or GitLab for version control and collaboration

II. QUANTIFYING SUCCESS

A. Court Detection

On the first video filmed with a smartphone, out of 40 court lines detected, 6 were wrong. That is an accuracy of 85%. All of those 6 sets of wrong court lines were flagged and the court lines from the previous frame were used instead. On the second video also filmed on a smartphone, out of 80 court lines detected, 9 were wrong (accuracy of 88%) but all correctly flagged as wrong and replaced with the previous frame's result. From Titcombe's report, the running time from comput ing average brightness to finding court lines' coordinates is 0.0467 ±0.0005s. This portion of code excluding computing average brightness (which takes virtually not time) is run every time the threshold for the white mask needs to be relaxed, which can be from 0 to 60 times (60 is an empirically chosen upper limit). Hence, finding the court lines for a single frame can take from about 0.0467s to about 2.802s

B. Ball Tracking & Hit Detection

We tested our algorithm on a video of a single tennis rally and the results are generated empirically. Occasionally, the algorithm fails to track the ball or hit but if anything is detected it is generally close to ground truth:

© 2025 IJIRCCE | Volume 13, Issue 4, April 2025 |DOI:10.15680/IJIRCCE.2025.1304247

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Bounce detection actual frame compared to detected frame											
	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th	11th
Actual	76	166	268	355	447	538	626	728	817	903	986
Detected	84	165	268	354	446	537	623	730	816	none	none

Hit detection actual frame compared to detected frame												
	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th	11th	12th
Actual	0	93	186	278	373	466	551	644	746	833	920	1000
Detected	none	93	186	278	373	466	none	644	747	833	920	1001

VI. ACKNOWLEDGMENTS

THIS REPORT OUTLINES THE PROPOSED TENNIS ANALYSIS SYSTEM, DESIGNED TO PROVIDE AF- FORDABLE AND ACCESSIBLE PERFORMANCE ANALYSIS FOR TENNIS PLAYERS. BY LEVERAGING THE YOLO OBJECT DETECTION MODEL, THE SYSTEM AIMS TO TRACK BOTH PLAYER MOVEMENTS AND BALL TRAJECTORIES IN REAL-TIME, OFFERING A DETAILED BREAKDOWN OF CRUCIAL MATCH MOMENTS. UNLIKE TRADITIONAL METHODS, WHICH CAN BE COSTLY OR MANUAL, THIS SYS- TEM WILL MAKE ADVANCED ANALYSIS TOOLS AVAILABLE TO AMATEUR PLAYERS, HELPING THEM IMPROVE THEIR GAME THROUGH OBJECTIVE DATA AND INSIGHTS.

REFERENCES

1. Omar Abdelaziz, Mohamed Shehata, and Mohamed Mohamed. Beyond tradi- tional single object tracking: A survey. *arXiv preprint arXiv:2405.10439*, 2024.
2. Matija Buri´c, Miran Pobar, and Marina Iva˘si´c-Kos. Adapting yolo network for ball and player detection. In *Proceedings of the 8th International Conference on Pattern Recognition Applications and Methods*, volume 1, pages 845–851, 2019.
3. Jing-Wei Liu, Ming-Hua Hsu, Chun-Liang Lai, and Sheng-K Wu. Using video analysis and artificial intelligence techniques to explore association rules and influence scenarios in elite table tennis matches. 2023.
4. Mohammed Gamal Ragab, Said Jadid Abdulkader, Amgad Muneer, Alawi Alqushaibi, Ebrahim Hamid Sumiea, Rizwan Qureshi, Safwan Mahmood Al- Selwi, and Hitham Alhussian. A comprehensive systematic review of yolo for medical object detection (2018 to 2023). *IEEE Access*, 2024.
5. N Gopika Rani, N Hema Priya, A Ahilan, and N Muthukumaran. Lv-yolo: Logistic vehicle speed detection and counting using deep learning based yolo network. *Signal, Image and Video Processing*, 18(10):7419–7429, 2024.
6. Jian Xiong, Liguo Lu, Hengbing Wang, Jie Yang, and Guan Gui. Object-level trajectories based fine-grained action recognition in visual iot applications. *IEEE Access*, 7:103629–103638, 2019.
7. Fei Yan, William J Christmas, and Josef Kittler. A tennis ball tracking algorithm for automatic annotation of tennis match. In *BMVC*, volume 2, pages 619–628. Citeseer, 2005.
8. Xi Yang and Haiming Wang. Design and implementation of intelligent analysis technology in sports video target and trajectory tracking algorithm. *Wireless Communications and Mobile Computing*, 2022(1):5633066, 2022.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi crossref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details