ISSN (Online): 3139-7689

Available online: www.ijsrmes.com

International Journal for Scientific Research in Modern Engineering and Science

Published in Conference Special Issue: Vol. 1, Special Issue 1, pp. 385-392, 2026

AI/ML-BASED TENNIS ANALYSIS SYSTEM

Udit Singh [1*]

[5]Department of Computer Science and Technology, Shri Shankaracharya Professional University, Bhilai, India.

Abstract

The advancement of intelligence and machine learning technologies has opened new opportunities for automated sports analysis. This paper presents a tennis analysis system that uses deep learning techniques. The system uses You Look Once (YOLO) object detection framework, convolutional neural networks for court key point extraction and PyTorch-based model training pipelines.

It. Tracks players and the tennis ball across video frames. The system extracts positional data of the court computes real-time player and ball speeds and generates an interactive mini-court visualization. The player detection module uses YOLOv8 for multi-object detection. A tuned YOLO variant is used for high-speed tennis ball tracking. The system uses a ResNet-50 backbone with a custom regression head trained on annotated court images to extract court points. These components are integrated into an analysis pipeline that processes raw match footage and produces structured perfor-mance metrics. Experimental results show detection accuracy across hard, clay and grass courts. The player detection achieves an average precision of 0.91.The ball detection achieves 0.84 average precision. The system accurately estimates player speed within a 5% margin of error. This is done using homography-based court-to-world coordinate transformations. This work contributes a modular and reproducible framework, for automated tennis match analysis. The framework can benefit coaches, analysts, broadcasters and players.

Keywords: *Tennis Analysis; YOLO; Pytorch; Computer Vision; Key Point Detection; Object Tracking; Player speed Detection; Deep Learning; CNN; Sports Analytics.*

** Corresponding author*

1. Introduction

Tennis is a tough sport that needs careful watching of how players move, where the ball goes and where they are on the court to get a good idea of how they are doing. Usually coaches and people who analyze the game have to watch every-thing by hand which takes a lot of time and can be unfair.. Now with the help of special computer programs that can see and understand things we can get a better and more accurate look at the game in real time.

New ways of finding things in videos like the YOLO program have made it much faster and more accurate to spot things. When we use this with computer programs that can look at important points and old computer vision techniques like homography we have a strong base to build a complete system to analyze tennis.

This paper is based on a YouTube video called 'Build an AI/ML Tennis Analysis System with YOLO, PyTorch and Key Point Extraction' by Abdullah Tarek. The video shows us how to make a system to analyze tennis videos including finding objects tracking the ball looking at points on the court and calculating speed. Our work adds theory, tests and talks about how to use this system in the real world.

The main things we did in this paper are:

- making a system that can analyze tennis videos from start, to finish
- making a YOLO program just for finding tennis balls even when it is hard to see them
- making a program that can find important points on the court very accurately
- finding a way to guess where the ball will go even when we cannot see it
- making a system that can measure speed accurately by mapping coordinates.

We think tennis analysis systems are very useful and our work can help make them better. Tennis analysis systems can help coaches and players understand the game of tennis.

1.1 Motivation

The need for information based on data in tennis has gotten really big for both amateur players. Big tournaments are using systems like HawkEye to help make calls on lines and to look at the path of the ball. These systems are still very expensive and most smaller clubs and amateur tennis players cannot afford them. A system that uses videos and is based on open-source machine learning frameworks can help fix this problem by making tennis analysis available to everyone, which is what people call democratizing high-quality tennis analytics [4]. Tennis players and tennis clubs can really benefit from tennis analytics. This new system can provide that. Tennis analytics is very important, for tennis players and tennis clubs.

1.2 Scope of the Study

This study focuses on monocular video analysis from a single broadcast-style camera viewpoint, which is the most com-monly available data source for tennis matches. The system is designed to operate on pre-recorded match footage without requiring specialized hardware beyond a standard GPU-equipped workstation. The analysis covers player detection and tracking, tennis ball detection, court key point extraction, speed estimation, and mini-court visualization.

2. Literature Review

The field of sports video analysis has gotten really big over the ten years. At first people used school computer vision tech-niques like background subtraction and color-based segmentation to find players and the ball. These methods worked well when everything was controlled but they did not do so great when the lighting was different or the court surface changed or the camera was moved around.

Then deep learning came along. Completely changed the way we do sports analysis. Convolutional neural networks were really good at learning features so they could find things. Classify them even when it was hard. The YOLO architecture that Redmon and others made was a deal because it could detect things in real time. It was fast and accurate because it treated detection like a problem. The new versions of YOLO. Like YOLOv3 and all the way up to YOLOv8. Have just got-ten better and better so now everyone uses them for real-time object detection tasks. Sports video analysis uses sports vid-eo analysis techniques, like these to get the job done.

For ball tracking the TrackNet system was created to track objects that are moving really fast in sports videos. TrackNet looks at frames in a row and makes a heatmap that shows where the ball is probably located which helps it track the ball even when it is moving so fast that it is blurry. Other people built on this idea and used things like attention and timing to make the tracking better.

There are a lot of ways to detect the court in a video. One way is to use something called homography, which matches up the coordinates of the image with the world coordinates of the court. This method is popular because it makes sense geo-metrically. Another way is to use learning, which is a kind of computer program that can learn things on its own. This method is better than the way of detecting lines because it is more robust. Someone named Kosolapov made a deep learn-ing system that's similar to TrackNet and it is really good, at detecting the important points on the court.

People have tried to figure out how fast things are moving in tennis using radar systems and using videos. The video meth-od involves tracking something as it moves across the screen calculating how far it moved in pixels and then using the size of the court and some math to figure out how far it really moved.

3. System Architecture

The proposed Tennis Analysis System (TAS) is organized as a modular pipeline consisting of five primary components: (1) Player Detection and Tracking, (2) Ball Detection and Tracking, (3) Court Key Point Extraction, (4) Speed and Metric Computation, and (5) Mini-Court Visualization

3.1 Player Detection and Tracking

The player detection is done using the YOLOv8 model. This YOLOv8 model was pretrained on the COCO dataset. The COCO dataset has a class for people. So the pretrained detector can find players without needing any training. The detector looks at each frame of the video on its own. It then finds the boxes around the people what class they are and how sure it is about each person it finds.

The ByteTrack tracking algorithm is used after the detector has done its job. This helps to keep track of the players from one frame to the next. This way the players always have the identity during the match.

The player IDs are looked at to see where they are, on the court. This helps to remove people who're not players like the ball boys, the referees and the people watching the game.

3.2 Tennis Ball Detection and Tracking

Detecting a tennis ball is really tough because it is small and moves fast. Sometimes it is even blocked from view. The regu-lar YOLO model that you can use away does not have a special category, for tennis balls and it does not work well for this job.

To make it better we took a YOLOv5 model. Adjusted it to work with a set of pictures of tennis balls that we got from vid-eos of tennis matches. We had 4,000 pictures of tennis balls in our set and these pictures showed different kinds of courts, lighting and how fast the ball was moving. We trained the model using a method that helps it learn from the pictures over time. We did this 100 times.Finding the tennis ball can be a little tricky because it is moving fast. Sometimes it is hidden.

We use a way to fill in the gaps when the ball is not seen so we can still figure out where it is going. We only accept the re-sults if the model is pretty sure it saw a tennis ball so we do not get answers. The model has to be least 60 percent sure it saw a tennis ball or we do not count it. This helps us get an idea of where the tennis ball is moving

3.3 Court Key Point Extraction

Getting the court key points right is really important for understanding the geometry and speed of the game. The court has 14 points where the service lines, baselines and sidelines meet and where the net is in the middle. We used a computer pro-gram to find these points from pictures of the court.

We used a kind of computer program called ResNet-50 to look at the pictures and find the important parts. Then we add-ed a custom part to the program that figures out the location of the 14 key points. The program was trained using a lot of pictures of courts 8,841 to be exact. It looked at the difference between what it thought the points were and what they really were.

After the program made its guesses we used some reliable computer vision techniques to make the guesses even better. For the points that're not, near the net we looked really closely at the area around each point and used special tools to find the lines and edges. Then we used a math formula to make sure all the points fit together perfectly like a puzzle. This helps us get an accurate picture of the court and the key points.

Table 1: Model Performance Metrics on Test Dataset

Component	Model	Metric	Value
Player Detection	YOLOv8	mAP@0.5	0.91
Ball Detection	YOLOv5 Fine-tuned	mAP@0.5	0.84
Court Key Points	ResNet-50 CNN	PCK@10px	0.89
Speed Estimation	Homography + Tracking	Error Margin	±5%

4. Methodology

4.1 Data Preparation

The tennis match videos were collected from broadcasts and tournament highlights. These videos were broken down into lots of parts like individual frames and they were looked at really closely. Each frame was taken apart. Labeled by hand using special tools that help us identify important parts of the image like where the tennis players are. To make sure the computer model works well in all kinds of situations the videos were changed in ways, such as flipping them sideways making them brighter or darker and adding some noise, to the picture. This was done to the tennis match videos so that the computer model can recognize things correctly when the match conditions are different.

4.2 YOLO Fine-Tuning for Ball Detection

The YOLOv5 model was adjusted using transfer learning it started with the weights it learned from COCO. When we trained the YOLOv5 model we used a settings. We used a batch size of 16. The learning rate was 0.01 at first. It got smaller over time using a cosine annealing schedule. The images were 640x640 pixels.
We split the data into three parts: 80% for training the YOLOv5 model, 10% for validation and 10%, for testing. The YOLOv5 model was trained on a NVIDIA RTX 3090 GPU. It took 80 epochs for the training to be complete.

4.3 CNN Key Point Model Training

The ResNet-50 model was made using PyTorch. We started with the backbone that had been trained on ImageNet. Then we made a custom part for the model to learn from scratch. When we trained the model we used the Adam optimizer. Started with a learning rate of 0.0001. The ResNet-50 models learning rate was reduced by half when the validation results stopped getting better. We used 32 pictures at a time. Trained the ResNet-50 model for 200 rounds. The ResNet-50 model got the pictures resized to 640x360 pixels before it looked at them.

4.4 Speed Estimation via Homography

The player and the ball speeds are figured out using a way to change coordinates. We have 14 points on the court that we know about. We use these points to make a matrix called H. This matrix helps us change the coordinates from the image to the court. We do this using a method called linear transform.
When we want to know how fast something is moving we look at how it has moved from one frame to the next. We do this in the world coordinates. Then we multiply this distance by how frames we see per second. This gives us the speed in meters per second. The player and the ball speeds are then changed into kilometers, per hour.Speed

$$(km/h) = |H \cdot p\_t{+}1 - H \cdot p\_t| \times FPS \times 3.6 \tag{1}$$

Where p_t and p_t+1 denote the pixel coordinates of the tracked object at consecutive frames t and t+1, H is the homography matrix, FPS is the frame rate, and 3.6 is the conversion factor from meters per second to kilometers per hour.

5. Results and Discussion

The system was tested on a set of 20 tennis match clips that were not used before. Each clip was between 2 and 5 minutes long. Showed tennis being played on all three main types of courts. Table 1 shows how well each part of the system worked.

The system used YOLOv8 to find the players. It worked really well. It was able to find the players 91 percent of the time even when the lighting and background were different. The system also used YOLOv5 to find the ball. It worked even bet-ter after it was fine-tuned. It was able to find the ball 84 percent of the time which is a big improvement over the original version that only found the ball 23 percent of the time. Sometimes the system missed the ball especially when it was being served fast over 200 km/h and the picture was blurry.

The system was also able to find the points on the court like the lines and the net. It was able to find these points 89 per-cent of the time which is better than the old way of doing it that only found them 72 percent of the time. When the system used a technique to refine its results it was able to find the points correctly 92 percent of the time as long as they were not near the net.

The system was also able to estimate how fast the ball was going when it was served. It was able to get the speed within about 4.8 percent, which is good enough for most tennis matches. The system also showed a version of the court with the players and ball on it which made it easy to see what was happening in the match. This visualization showed the positions of the tennis players and the ball, in time which helped people understand what was going on in the tennis match. The ten-nis players and the ball were shown on the mini-court. It looked like they were really moving around on the tennis court.

5.1 Limitations

Several limitations were found during evaluation.

One limitation is that the system is not good at detecting balls during serves. This suggests that we need to look at frames over time like TrackNet does.

* The model that finds points on the court has trouble when the court is partially covered by players or shadows.
* Another issue is that the system assumes the camera angle is always the same but if the camera moves or zooms the system would need to be adjusted or re-calibrated.

We plan to work on these limitations in the future by using architectures that consider frames and, by making the camera calibration adaptive.

6. Implementation Details

The system is made using Python 3.10 and some tools like PyTorch 2.0 Ultralytics YOLOv8, OpenCV 4.8 and NumPy. It uses OpenCV to read video then it looks at each part of the video. Uses the detection and tracking parts and finally it writes the video with notes using VideoWriter. The whole system works at 12 frames per second on an NVIDIA RTX 3090 GPU which is good enough to look at video that is not live.

The project is set up in a way that's easy to understand with separate parts for finding the ball finding the players finding important points on the court calculating speed and showing the video. This makes it easy to change or upgrade any of these parts if better models come out. You can find the code and the pre trained model weights, on GitHub, which means people can use it and add to it [13].

7. Conclusion

This paper is a tennis analysis system that uses artificial intelligence and machine learning. It uses something called YOLOv8 to detect the players, a version of YOLOv5 to detect the ball and a type of neural network called ResNet-50 to find important points on the court. It also has a way to estimate how fast the players are moving.

The system works well and can be used for a lot of things. It can help coaches and players see how well they are doing it can help with planning strategies it can make TV broadcasts more interesting. It can even give players feedback on how to improve. This system is also good because it uses open-source tools and data that anyone can get. It is cheaper than some other systems like HawkEye.

In the future the people who made this system want to make it even better. They want to use ways to track the ball when it is moving really fast they want to use more than one camera to get more information and they want to be able to under-stand more about what is happening in the game. They also think that using something called transformer-based vision models could help them find points on the court more accurately. The tennis analysis system will keep getting better and better. It will be able to do more things to help players and coaches. The system is, about tennis analysis. It uses tennis analysis to make the game more fun and interesting

References

[1] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-444.

[2] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE CVPR (pp. 779-788).

[3] Tarek, A. (2023). Build an AI/ML Tennis Analysis System with YOLO, PyTorch, and Key Point Extraction [Video]. YouTube. https://youtu.be/L23oIHZE14w

[4] Beal, R., Norman, T. J., & Ramchurn, S. D. (2019). Artificial intelligence for team sports: A survey. Knowledge Engineering Review, 34.

[5] Pingali, G., Jean, Y., & Carlbom, I. (1998). Real time tracking for enhanced tennis broadcasts. In Proceedings of the IEEE CVPR (pp. 260-265).

[6] Redmon, J., & Farhadi, A. (2018). YOLOv3: An incremental improvement. arXiv preprint arXiv:1804.02767.

[7] Huang, Y.-C., Liao, I.-N., Chen, C.-H., et al. (2019). TrackNet: A deep learning network for tracking high-speed and tiny objects in sports applications. arXiv:1907.03698.

[8] Naik, B., Hashmi, M. F., & Bokde, N. D. (2022). A comprehensive review of computer vision in sports: Open issues, future trends, and research directions. Applied Sciences, 12(9), 4429.

[9] Farin, D., Krabbe, S., De With, P. H. N., & Effelsberg, W. (2003). Robust camera calibration for sport videos using court models. In Proceedings of SPIE Storage and Retrieval Methods for Media Databases.

[10] Kosolapov, S. (2023). Tennis analysis using deep learning and machine learning. Medium. https://medium.com/@kosolapov.aetp/tennis-analysis-using-deep-learning-and-machine-learning

[11] Vignolo, L. D., Milone, D. H., & Scharcanski, J. (2018). Feature-based ball detection in soccer videos under complex conditions. Neurocomputing, 286, 118-125.

[12] Hartley, R., & Zisserman, A. (2003). Multiple View Geometry in Computer Vision (2nd ed.). Cambridge University Press.

[13] Tarek, A. (2023). tennis_analysis [GitHub repository]. https://github.com/abdullahtarek/tennis_analysis