Hoops Radar: Player Tracking with NBA Broadcast footage

Tomas Coghlan
Stanford Department of Computer Science
tcoghlan@stanford.edu

Abstract

Tracking basketball players using publicly available broadcast footage poses a unique computer vision challenge due to partial occlusions, angled perspectives, and limited field of view. I introduce HoopsRadar, a system that combines multiple YOLOv8 models with homography transformations to map ball and player positions from NBA broadcast videos onto a to-scale, two-dimensional court representation. Our approach detects players, the ball, and court markings independently, using a custom-trained model for each, and fuses their outputs via geometric mapping. To enhance temporal coherence and handle occlusions, we integrate ByteTrack, a tracking-by-detection framework with low-confidence association and Kalman filtering. This system paves the way for downstream applications such as play classification, spacing evaluation, and player influence analysis, even for researchers without access to league-provided tracking data.

1. Introduction

Scouts in sports analyze hours of footage for player evaluation and strategy purposes. However, computer vision models which are able to collect data from actions on the court may generate valuable conclusions that the human eye can miss. The use of computer vision in sports has grown substantially in the last decade. In the United States, professional leagues for baseball, basketball, and football have all installed tracking technology and cameras in their stadiums to generate more data for evaluating players. In basketball specifically, many NBA teams use several cameras positioned in stadiums or a fish eye view camera which captures the whole court at once. This improves the accuracy of computer vision tracking techniques. However, much of this footage and data is not publicly available due to analytics competition between teams. The goal of this project is to train models which can provide accurate and useful data from easily available broadcast footage.

A model which tracks player and ball movement across the court also could have applications in player evaluation and entertainment media. Certain clips could be classified under play types and accessed by search. Players could be evaluated not just by end-of-play statistics, but by the effect they are having with their movement on the court. Specifically, a model capable of using broadcast footage for tracking would be a significant resource for sports analysis for researchers outside of the industry, since much of the tracking data collected by teams is unavailable.

In this project, I fine-tuned separate YOLOv8 models for ball, player, and court detection. Each of these models was used in conjunction to map the ball and player locations of each frame of broadcast footage to a 2-D representation of the basketball court.

2. Related Work

Previous attempts at ball/player tracking Difficulties usually arise with using broadcast footage for computer vision tracking since the camera does not include the entire court and shows the court from the side at an angle. Francia [1] used CNNs to classify basketball actions, however the use of YOLO models combined with homography in Pandya et al. [2] seemed much more promising. However, this was successful in American football, where the field is much larger, contains far more identifiable features for homography transformations, and public location data is available for validation.

One main issue with using broadcast footage is that scaling the locations on the court to a flat mapping is difficult from just one angle. To do this effectively, homography transformations on pixels are broadly used. The transformation matrix is created using at least shared 4 pixel locations between the two images (the broadcast image and the flat map). Large, common court/field markings are often used as the pixel locations for creating this homography matrix, as they can be easily accurately. However, if these markings are blocked by camera, player, referee, or fan movement, the homography matrix cannot be calculated for that frame. In Wen et al., [4] blocked locations of these markings are borrowed/averaged out from nearby frames, creating a smooth indication of where the markings are located

in the image. This idea was crucial in my implementation and application of homography transformations.

3. Methods

The HoopsRadar model structure (Figure 1) contains 3 separate YOLOv8 models for ball tracking, player tracking, and court markings tracking (corners of the court, corners of the paint, etc.). These models generate the bounding boxes and tracking IDs for each object in an image.

Homography transformations with the location of the court markings were then used to map the players onto a to-scale, 2-D diagram of an NBA court from a bird's eye view. The tracker IDs of the players and the location of the ball was used to determine who was the ball handler in the image.

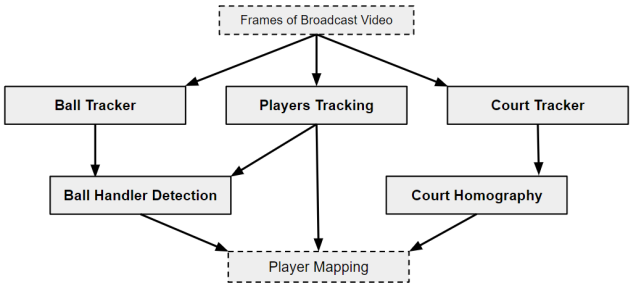

Figure 1. HoopsRadar Model Strucure

3.1. YOLO

YOLO (You Only Look Once) [3] is a real-time object detection system that aims to detect and classify objects within an image in a single forward pass through a convolutional network. YOLO divides the input image into a grid and uses regression to predict bounding boxes and class probabilities for each grid cell. It then uses non-maximum suppression to filter out duplicate detections. The non-maximum suppression is set by the IOU (intersection over union) hyperparameter, which controls how much proportion of overlapping area is allowed by multiple detections. YOLO has fast inference times, which is vital in our application where we apply inference to every frame in a video. YOLO also is better at learning more general representations of objects and predicts fewer false positives, which is preferred over false negatives in my context.

3.1.1 ByteTrack

The results of the player model is passed into the ByteTrack [5] model. The ByteTrack model provides the tracking IDs for each detection in each frame. The ByteTrack uses a Kalman filter, a recursive algorithm used to estimate the state of a linear system from noisy observations. It is widely applied in object tracking to predict and update an object's position over time, accounting for uncertainty. ByteTrack differs through a process that assigns IDs to detections with both high confidence scores and low confidence scores so that occluded objects are not ignored. This is good for our application which includes many instances of players overlapping each other.

3.2. Video Mapping

3.2.1 Court Homography

A homography transformation describes the relationship between two images of the same planar surface from different perspectives. A homography matrix maps points in one image to corresponding points in another image. In our case, the two images are the broadcast view of the court and the our 2-D map. Below we see how the homography matrix is used to calculate the new point:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = H \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{1}$$

To calculate the homography matrix accurately, a minimum of four pairs of non-collinear points is required. This is because the homography matrix has eight degrees of freedom (8-parameters) that need to be determined to represent the transformation between two images.

$$\begin{cases} u = h_{11}x + h_{12}y + h_{13} \\ v = h_{21}x + h_{22}y + h_{23} \end{cases} \tag{2}$$

The issue here is that finding 4 non-collinear points can be difficult since many of the easily detectable court markings lie on the same lines. It means that the homography transformation is not easily to calculate in many frames of an broadcast clip.

The center of the bounding boxes from the court markings model were matched with corresponding points on a to-scale, 2-D diagram of an NBA court from a bird's eye view. These points are used to create the homography matrix.

The bounding boxes for each of the players was used to find the points on the broadcast footage that represent their relative positions. The x-coordinate of the player was the mean value of the two sides of the bounding box. The y-coordinate was chosen as the bottom of the bounding box plus 5% of the total height of the bounding box. This generally produced results close to each players' feet. In frames where a player disappears, the locations of that player from surrounding frames is used. In addition, the direction/momentum of player was used to determine the most likely location.

3.2.2 Ball Handler Detection

To find the player who was the ball handler, we check if the center of the bounding box for the ball detection was inside the bounding box of any of the player detections for the current frame and for the 10 frames on either side of the current frame. If the player detection with the same tracker ID was the most common result, it was marked as the ball handler for that frame. Otherwise, no ball handler was marked.

4. Dataset and Features

All datasets were collected from Roboflow.

For ball detection, a combination of two different ball detection dataset were used to provide better model generalization (https://universe.roboflow.com/basketballdetector/nba_dataset, https://universe.roboflow.com/gaga-lala-7qi2v/basketball-ball-1ddrw). These datasets provided bounding box labels for basketballs in broadcast footage. This provided 7529 labeled examples to train on.

For player detection, a single Roboflow dataset was used. (https://universe.roboflow.com/betatracker/nba-players-ronfv). This dataset provided bounding box labels for the players and referees in broadcast footage. This provided 1968 labeled examples to train on.

For Court detection, a single Roboflow dataset was used. (https://universe.roboflow.com/betatracker/nba-court). This dataset provided bounding box labels for the baseline markers and the corners of the "paint" in broadcast footage. This provided 676 labeled examples to train on.

Each of these datasets were enhanced with augmented data examples, including rotations and greyscale.

5. Results

The accuracy and precision of the ball, player, and court detection models were evaluated on validation datasets from Roboflow data:

Table 1: Results of YOLO models on seperate validation sets

Model	Epochs Trained	Val Images	Box Precision	Box Recall
Ball Detection	10	1310	0.925	0.831
Player Detection	30	186	0.972	0.956
Court Detection	53	189	0.958	0.934

Figure 2. Results

A precision above 92% was achieved for each of the detection models. Recall on the ball detection model suffered in comparison to high recall scores from the player and court detection models.

As is seen in figures 3, 4, and 5, the model can effectively plot the locations (shown in red pixels) of the players' locations to a 2-D visualization using homography. The original broadcast footage is transformed to match the scale of our 2-D map.

Figure 3. Example of Broadcast image frame

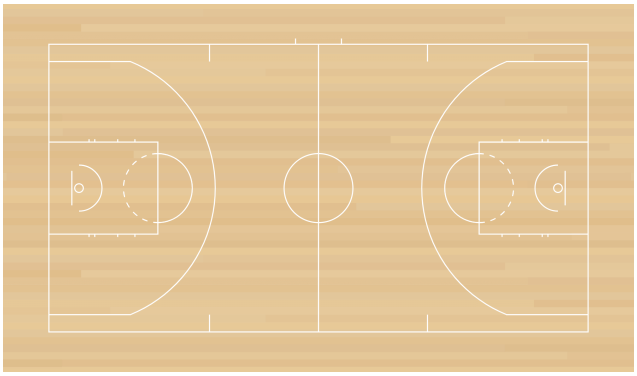

Figure 4. Our 2-D Map of the court (to-scale)

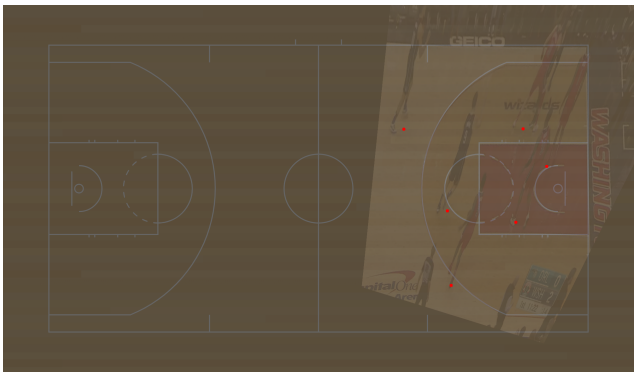

Figure 5. Example of Homography mapping of Figure 4 onto Figure 3

As is seen in the confusion matrices on the validation data for each model (Figures 6, 7, 8), the models were largely successful. However, two limitations standout. In the ball detection model, we see that 13% of the basketballs in the broadcast frames were not detected. So while the ball model is very good at avoiding false positives, it has the tendency to of false positives in the data

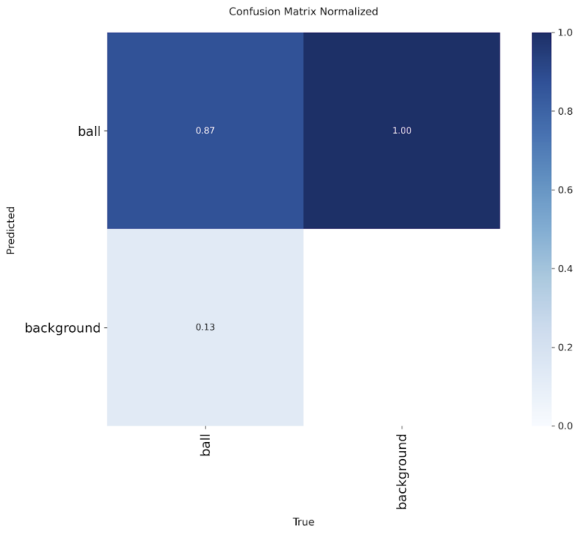

Figure 6. Confusion Matrix for Ball Classification

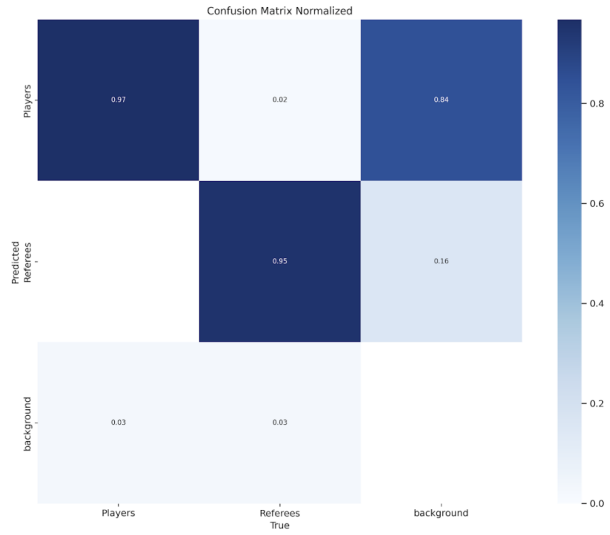

Figure 7. Confusion Matrix for Player Classification

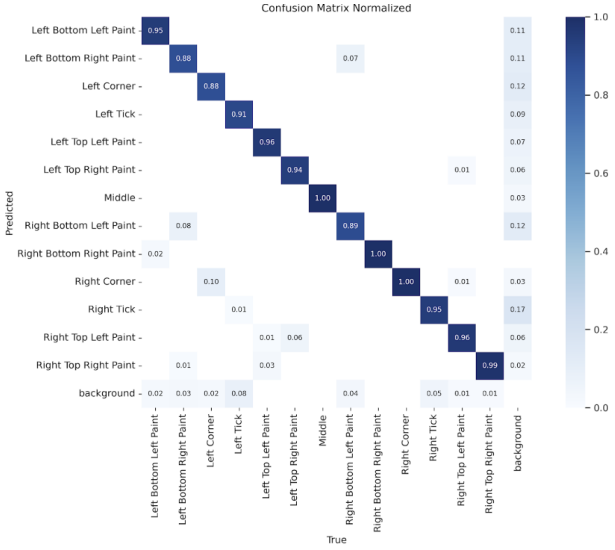

Figure 8. Confusion Matrix for Court Markings Classification

6. Conclusion and Future Work

The HoopsRadar model performed fairly well at tracking player positions across the court and at generating visualizations of the tracking on a flat 2-D mapping of the court. The opens up many downstream tasks that could be vital in player evaluation. The positions of the player with the ball and the defenders could be used to judge the difficulty of the shot.f the shot. Playcalls/strategies could be classified based on player movement. Position can also be used to calculate player velocity, spacing, and defensive presence on the court.

Future improvements on the object detection models themselves could include (if made available) evaluating location data against official NBA data. This data could also be used to determine the most accurate pixel used in the players' bounding boxes to be used in homograph transformations. The use of a RNN or transformer could also be considered, information from neighboring/previous frames could inform the location of the players during moments of overlap. This could also help with classifiying specific players on the court based on their jerseys or physical characteristics.

References

[1] S. Francia. *Classificazione di Azioni Cestistiche mediante Tecniche di Deep Learning*. PhD thesis, 04 2018. 1

[2] Y. Pandya, K. Nandy, and S. Agarwal. Homography based player identification in live sports. pages 5209–5218, 06 2023. 1

[3] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection, 2016. 2

[4] P.-C. Wen, W.-C. Cheng, Y.-S. Wang, H.-K. Chu, N. C. Tang, and H.-Y. M. Liao. Court reconstruction for camera calibra-

tion in broadcast basketball videos. *IEEE Transactions on Visualization and Computer Graphics*, 22(5):1517–1526, 2016. 1

[5] Y. Zhang, P. Sun, Y. Jiang, D. Yu, F. Weng, Z. Yuan, P. Luo, W. Liu, and X. Wang. Bytetrack: Multi-object tracking by associating every detection box, 2022. 2