Video-Based Basketball Shooting Training System

Mi Guanyu
The Chinese University of Hong Kong, Shenzhen
120090700@link.cuhk.edu.cn

Cheng Junzhi
The Chinese University of Hong Kong, Shenzhen
120090777@link.cuhk.edu.cn

Abstract

This project works on a Video-Based Basketball Shooting Training System that can generate the shooting heat map of the trainer showing the hit rate of the player in different areas. It mainly consists of three parts: shot event recognition, shot result justification, and player localization. All these three tasks are based on the detection algorithm for player, basketball, hoop and court. Previous works have shown that YOLO is a good choice for real-time detection, the focusing-on-hoop-area algorithm may be an excellent way to do the shot result justification, and the homography transform is excellent for player localization. Inspired by the previous work, we propose the shot event recognition model using the position of the player and basketball, the shot result justification model based on similarity, and the player localization model based on homography transform. In the experiment, our group explored the feasibility of the combination of detecting model and tracker and determine the threshold value estimating the goal. The methodologies are clarified as workable for the sample videos.

1 Introduction

Basketball is one of the most popular sports in China, and it is all about getting the ball into the hoop. This makes shooting an essential technique in basketball. Pro players have shooting coaches who help record their training data and improve their shooting motion. However, the amateur player does not have the condition to train this way. The most common method amateur players use is recording their shooting training and analyzing their shooting motion via video. A camera-based shooting training model that can record the hit rate of the player in different areas is needed by many amateur players.

Three tasks must be solved to conduct such a training system: shot event recognition, shot result justification, and player localization. These three problems are all based on player detection, basketball detection, hoop detection, and court detection. The focus of the question is how to achieve an accurate detection model and how to use the detection result to accomplish these three tasks.

In this paper, we propose a Camera-Based Real-Time Basketball Shooting Training System that can generate the shooting heat map of the trainer showing the hit rate of the player in different areas. Figure 1 shows the main flow chart of our algorithm.

1.1 Related Work

Detection
Cheshire *et al.* [1] uses HOG and SVM to detect the player in their player tracking system. However, this method has a 70% missing rate and has to additionally use a color-based detector and classifier to

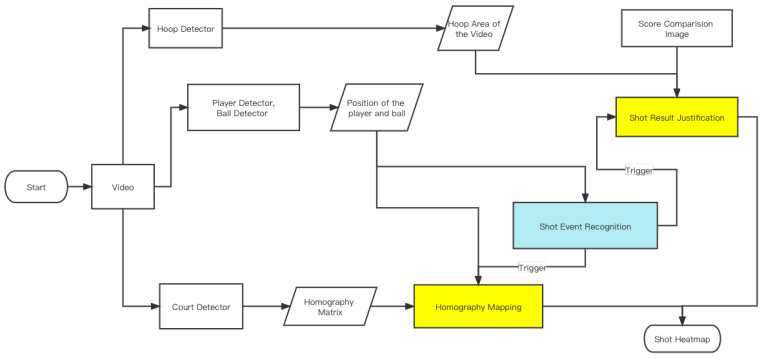


Figure 1: Main Flow Chart of Algorithm

help increase the accuracy. As for court detection, Cheshire *et al.* [1] first uses a canny edge detector to find the edges and then uses the Hough transform to detect the court area. Though this only works for the full-court video, starting with the edges from the canny edge detector is an excellent way to conduct court detection. Wen *et al.* [4] uses dominant color detection to detect the court. This method is suitable for court color that is different from the background. Fu *et al.* [2] uses YOLO to detect the hoop. This method has high accuracy and can do real-time detection.

Shot Result Justification
Wen *et al.* [4] uses frame difference to detect motion in the hoop area of the video to justify the shot result and has high accuracy. Focusing-on-hoop-area algorithm is an excellent way to do the shot result justification.

Player's Localization
Both Cheshire *et al.* [1] and Wen *et al.* [4] both use homography transforms to localize the player's position on the 2D basketball court. They both use four corners of the court to find the homography matrix. For this project, we focus on half-court and even areas inside the three-point line, a new way to find the homography matrix is needed.

Shot Event Recognition
Toshev *et al.* [3] propose an open-source human pose detection, including shot detection. However, it has a complex network structure and needs GPU to conduct the inference step. A more straightforward method is needed for this project.

1.2 Contribution

This project pioneered the creation of a camera-based, real-time shooting training system. The main contribution of this work is the design of the overall pipeline, the proposed shot event recognition model based on object detection, the shot result justification model based on similarity, and court detection based on the canny edge detector and contour detector.

2 The Proposed Algorithm

2.1 Algorithm Overview

As is shown in Figure 1, given an input video, the player detector and ball detector will return the position of the player and ball. Based on the position, the Shot Event Recognition model can detect whether a shot event happens. Every time a shot event happens, it will trigger the player localization model and shot result justification model.

For the player localization model, the court detector will help compute the homography matrix, and the player localization model will use this matrix to localize the player on the 2D court.

The Shot Result Justification model will use a hoop detector to focus on the hoop area of the video. Then it will compute the similarity of the hoop area image with a comparison image (a hoop with a

basketball in it). If the similarity is bigger than a threshold (for example, 0.9), the model recognizes it as a hit.

With these three models, this algorithm can generate a real-time shot heatmap.

2.2 Player Detection, Basketball Detection, Hoop Detecrtion

YOLOv4 model is a suitable model for the project, which is more efficient than Faster RCNN model architecture. Figure 2 made by jiangdabai shows the structure of YOLOv4.

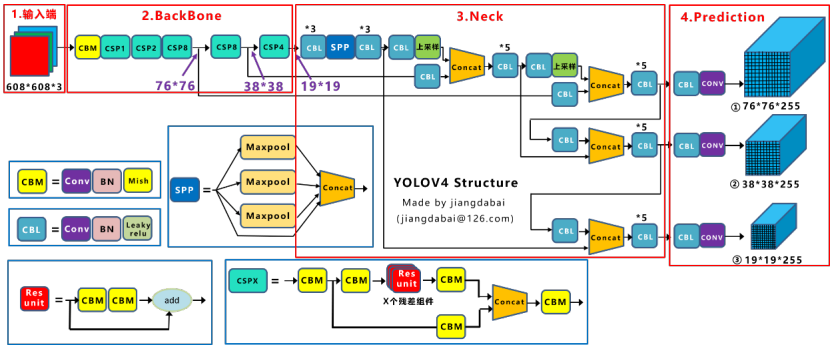


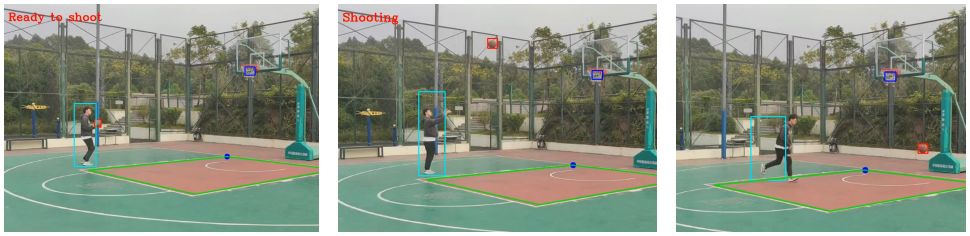
Figure 2: YOLOv4 Structure

By using YOLOv4, we are able to detect tiny objects faster and more precisely. Since there are not many ready-made weights for detecting all the objects at once, YOLOv4 pretrained weight used in the project AI Basketball Games Video Editor (https://github.com/OwlTing/AI_basketball_games_video_editor) was used by this project.

Because the detecting will cost about 1.5s a frame on average, Yolov4-pytorch model was used to detect components for every ten frames to decrease the processing time. In additional frames that do not use model to detect, CSRT tracker was implemented to find balls since balls can have large position changes and might effect the result of event recognition. CSRT tracker was also implemented to prevent balls from not being detected in the detecting frames.

2.3 Shot Event Recognition

Using the position of the player and ball detected by the player detector and basketball detector, we create recognition rule like this, which was shown in Figure 3:



(a) (b) (c)

Figure 3: Shooting Event Recognition Example

When the ball is around the player, the situation will be recognized as “ready” When the ball is above the head of the player for 1/3 of the player’s height and the situation is “ready”, the shooting begins and the goal justification begins. When the ball is at the height of 1/2 of the player’s height, the shooting ends.

2.4 Shot Result Justification

First, we extract a picture of the hoop at the time of the shooting begin as the comparison image. Once the situation is "shooting", this work compares the similarity of the hoop area and the comparison image for every frame as show in Figure 4.

(a) (b)

Figure 4: Similarity Compares Between Two Images

To make the justification more robust and make the threshold value as equal as possible for different videos, we tested different algorithms to compute similarity including Mean Squared Error and Structural Similarity Index.

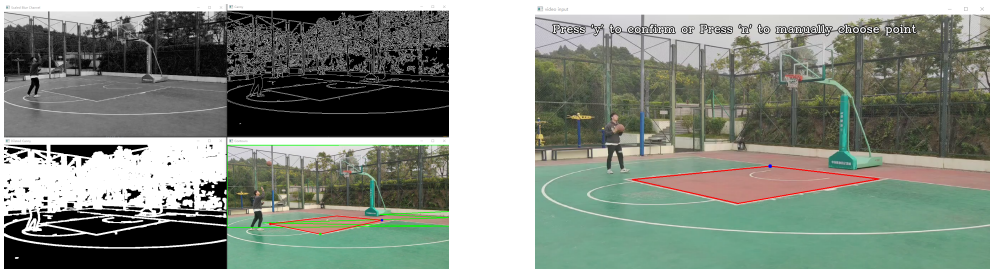
$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2$$

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

However, we finally chose to directly compare the mean difference between two images in the blue channel. The experiment will be discussed deeper in the later section.

2.5 Court Detection

This happens on the first frame of the video. First, the blue channel of the image is extracted to make the line brighter. Then canny edge detector is used to find the edges in the frame. After that, we dilate the image to reduce the discontinuity of edges. In the end, a contour detection is taken to find the approximate quadrilateral. We will use the four vertexes of the quadrilateral to help calculate the homography matrix in 2.6.



(a) (b)

Figure 5: Court Detection

While the camera was not fixed during recording the video, the quadrilateral and the hoop binds together by fixing the relative position after the first frame. Therefore the quadrilateral can be in a relative accurate position to implement a correct homography transform.

2.6 Player Localization

The player localization mainly based on homography transform. Homography transform is a transform that takes an image of an object as input and transform it into the image in other view of the same object. Homography transform is conducted by using the formula:

$$\begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x_s \\ y_s \\ 1 \end{bmatrix} \tag{1}$$

The three-by-three matrix is the homography matrix. Scale h_{33} to 1, it has in total 8 degree of freedom. In total it takes four points to calculate the homography. Using the vertexes found in 2.5, now we have the homography matrix. Player detector will return a bounding box for the player. We use the mid-point of the bottom line of the bounding box to denote the position of the player on the basketball field. Denote this point as P. Every time a shot is detected, the algorithm will perform homography transform on the frame immediately and the position of P on the transformed image is the position of the player on the 2D basketball field.

3 Experiments

3.1 Object Detection

This project detects people and hoop based on the YOLOv4 model, while the detection of the ball is implemented by the combination of YOLO detecting and CSRT tracking method. Since hoop is practically stationary in the video and the player is easy to recognize, the accuracy of their detection is about a hundred percent. Nevertheless, the basketball in the video is a tiny and fast object which is difficult to detect. Therefore, in the experiment we artifically count the time the box is at the right position and compute the accuracy to see whether the methodology is appropriate.

Table 1: Quantitative experimental results of basketball detection

Video	Total Length(s)	Right Position Time(s)	FPS	Accuracy(%)
1	20	17	30	85.00
2	43	42	30	97.67
3	49	35	30	71.43
4	23	23	30	100.0
Total	135	117	30	86.67

We can see from Table 1 that the accuracies of sample videos differs, and such accuracies is related to how large the ball is in the video. In video3, the camera is distant from the ball so it has lower detection accuracy. For other sample video at an appropriate distance, the accuracy is acceptable.

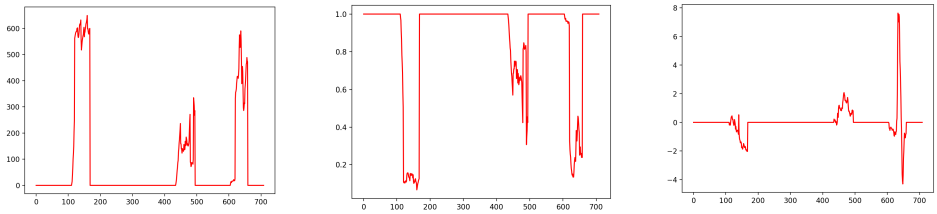
3.2 Goal Event Justification

In the experiment, we record the similarity shift using different algorithms and plot them. Consequently, we can choose the algorithm that can obviously distinguish whether the ball goals at a certain threshold value for all video. The similarity compared using MSE is shown in Figure 6(a), SSIM shown in Figure 6(b), Mean Subtraction shown in Figure 6(c). In the sample video used, the first and second shots did not goal, and the last one goaled.

We can find that Mean Subtraction has the most significant difference between goal and nongoal.Then we compare between different videos to decide a threshold value, as shown in Figure 7

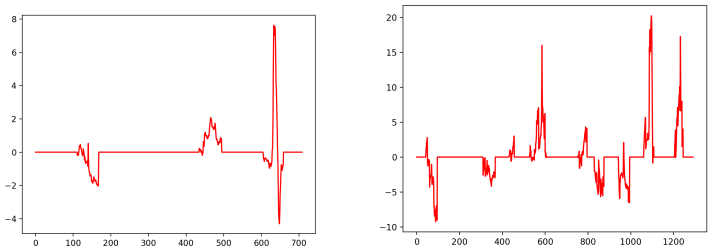
It is shown that in the first video the highest value is 7.6, which is considerably higher than the nongoal value in the second video, thus erroneous judgment is unlikely to happen. As we have decided the threshold value as 7.5, we artificially check whether there are misjudgment of goal. The result is shown in Table 2.

Since the reason for the misjudgment mainly comes from ball detection and shooting recognision, sample videos emphasized that our goal-justified methodology is suitable.



(a) (b) (c)

Figure 6: Similarity Change Using Different Algorithm



(a) (b)

Figure 7: Similarity Change in Different Sample Videos

4 Discussion and Conclusion

This project successfully proposed a camera-based shooting training model that can generate a shooting heatmap. Experiments have shown that the accuracy of our algorithm is satisfying. However, further improvement is still needed.

The accuracy of our algorithm mainly depends on the accuracy of the detection algorithm. YOLO, at most time, works perfectly. However, when the resolution becomes too low or when the ball is too far away from the camera, it cannot detect the ball. Further study on ball detection is needed to reduce the miss rate.

Limited by computing resources (no GPU), this project only uses the relationship between the player and basketball to do the shot event recognition. However, this can lead to a false positive when the player throws the ball in the air but does not shoot, which is common in shooting training. More advanced action recognition model like human pose recognition model is needed to solve this problem.

With success in the half-court training, the training area can be extended from half-court to full-court in the future. Moreover, Shooting trajectory analysis and shooting pose analysis can be included in the future to provide users with more information about their shooting.

Table 2: Misjudgment Count

Video	Shots in Video	Shots Detected	Goal in Video	Goal Detected	Misjudgment
1	3	3	1	1	None
2	7	9	3	3	2 Shot Rec
3	6	4	1	0	2 Ball Detect
4	3	3	1	1	None

References

[1] E. Cheshire, C. Halasz, and J. K. Perin. Player tracking and analysis of basketball plays. In *European Conference of Computer Vision*, 2013.

[2] X.-B. Fu, S.-L. Yue, and D.-Y. Pan. Camera-based basketball scoring detection using convolutional neural network. *International Journal of Automation and Computing*, 18(2):266–276, 2021.

[3] A. Toshev and C. Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1653–1660, 2014.

[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 calibration in broadcast basketball videos. *IEEE transactions on visualization and computer graphics*, 22(5):1517–1526, 2015.