Basketball Video Analysis for Automated Game Data Acquisition Deep Learning

Diego Rodriguez Garcia, Xinrui Yu, and Jafar Saniie
Embedded Computing and Signal Processing (ECASP) Research Laboratory (http://ecasp.ece.iit.edu/)
Department of Electrical and Computer Engineering
Illinois Institute of Technology, Chicago IL, U.S.A

***Abstract*— The utilization of data analytics to gain insights into the game of basketball has seen a remarkable surge in the past decade. Leagues such as the National Basketball Association are continuously exploring innovative methods to analyze game data, an approach that has significantly influenced the dynamics of the game. But to perform these analyses, a growing amount of data is needed, which is traditionally annotated by humans. This work proposes a 3-stage system able to automatically acquire relevant basketball game data from a broadcast video. The first stage is an object detector combined with a tracking algorithm to extract the main elements present in a basketball game video. Then, the players' visual information is analyzed to identify the players based on pixel color analysis and number recognition. Finally, a statistics generation algorithm assigns the game events to the corresponding player and team, so that the system can be used as an aid for box score annotation in major leagues, low-cost annotation in amateur games, or in-depth game video analysis.**

Keywords*— *Sports Video Analysis, Deep Learning, Object Detector, YOLOv8, Object Tracking,

I. INTRODUCTION

In recent years, the analysis of data in many industries and sectors has completely changed how companies, organizations, and teams approach the challenges they face daily. In professional sports and the entertainment industry, this data revolution kicked off in the first years of the 21st century when Oaklands Athletics' general manager Billy Beane adopted the use of deeper data analytics in their decision-making strategies [1]. Since then, many leagues and sports associations have followed this path and these data-driven techniques have kept growing and evolving with the introduction of Machine Learning (ML), Deep Learning (DL), and predictive analysis.

The National Basketball Association (NBA) is one of those leagues that has completely changed with the advent of data analysis. Since the late 2000s, all franchises in the league have shifted toward the use of data to make better more informed decisions in all aspects of the organization's operation. This completely revolutionized how the game was understood and had a huge impact on how the game was played.

However, the analysis that leads to these results and impacts on the game requires a vast amount of data which is traditionally annotated by humans, in a slow and costly effort. Furthermore, more advanced statistics, such as shot location, contested or uncontested shot, catch-and-shoot, or pull-up jumpers, require much more information to be annotated compared to the traditional points, assists, and rebound statistics.

II. OBJECTIVES

The goal of this paper is to develop a system referred to as Basketball Video Analysis (BVA) able to automate the acquisition of relevant data in basketball game videos for analysis and create a framework to ease the acquisition of new data with future developments. This paper aims to recognize some of the game actions using the broadcast video of it. To be able to detect these events, certain classes in the video have to be detected and tracked, these classes can be referred to as primary classes: players, ballhandler, ball, basket, and made basket.

After detecting these objects in the video and establishing relationships between them, an immense amount of data can be acquired. However, in this paper, the goal is limited to obtaining the following statistics from the game: passes, field goal attempts, field goals made, and rebounds. To correctly assign these statistics to the corresponding team and player, the system must identify each of the players on the court and determine to which team they belong. The output of this system would be an annotated video with the primary objects detected and the mentioned stats for each team and players identified on the court.

III. SYSTEM DESIGN

The BVA receives the game video as input and must be able to detect and track the primary classes mentioned, post-process some of these classes (players and ball-handler), and finally organize, manage, and establish relationships between the objects detected to obtain the stats defined in the scope. The BVA can be divided into 3 phases or stages (Fig. 1).

A. Stage 1: Object Detection and Tracking

In this stage, the video is fed to an object detection algorithm to detect and track the primary classes. This stage outputs the class, confidence, bounding box, and tracking ID of the objects detected.

1) Object Detection: YOLOv8.

Object detection involves recognizing that certain objects of interest are present in an image and estimating the boundaries of those objects (bounding box) as well as the class to which they belong (classification confidence).

For this paper, as the software could be used for real-time applications, a One-Stage Object Detector would be better suited (generally faster in inference time). Therefore, it has been decided to use one of the most widely known object

detectors, You Only Look Once (YOLO) version 8 by *Ultralytics*, the last version of the model first developed by Redmon et al. [2].

This model outperforms the previous YOLO versions. There are 5 different YOLOv8 models depending on the number of parameters, where the smaller the model the faster it is and the higher amount of FPS can be achieved during inference, but the accuracy is worse. This imposes a trade-off between a faster system and a more accurate one. Given the requirements of the BVA, an accurate model is needed, but as the real-time application could be used in some situations, the large model was selected as it is the one with the best balance between inference time and accuracy.

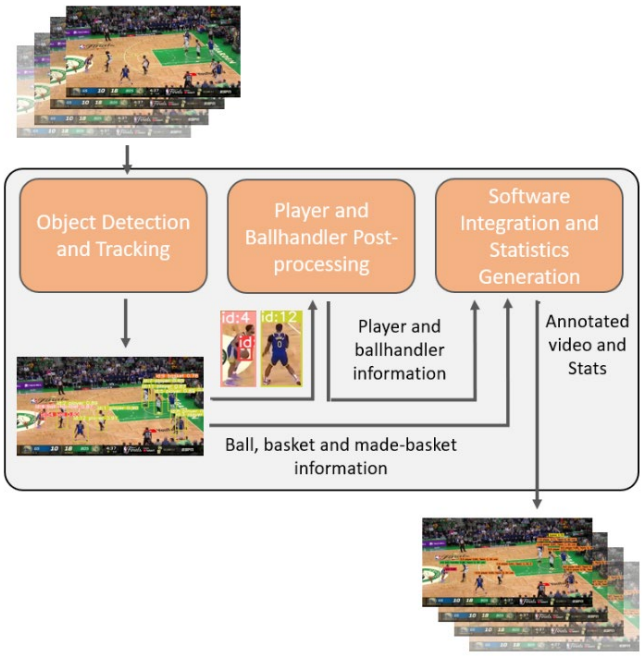

Fig. 1. Basketball Video Analysis Architecture.

2) Object Tracking: BoT-SORT

Object tracking is another relevant task in Computer Vision. The goal of this task is to, given a set of objects detected in a frame (using an object detector such as YOLO), determine if the detected objects correspond to instances of those same objects in the previous frames of a video.

Object tracking is fundamental for this paper for several reasons. First of all, to extract the maximum amount of data, all the players on the court should be identified at all times, so that when an event occurs, it can be assigned to a specific player. As will be shown in the next sections, the identification process is the most challenging part of this system, and having to execute this process in every single frame of the video is very time-consuming. Secondly, there are numerous situations where, due to the algorithms and models applied, the identification of players and the ballhandler is virtually impossible. This is why adding a tracking algorithm can help in event assignment and significantly reduce the time-consuming operations of player identification.

Multiple algorithms are continuously being improved and tested on the Multiple Object Tracking Benchmark (MOT) [3]. For the BVA, 3 object trackers were considered: DeepSORT [4], ByteTrack [5], and BoT-SORT [6]

After testing these 3 object-tracking algorithms and comparing their performance on different benchmarks, Bot-SORT was selected due to its great performance, especially in situations with occlusion.

3) Dataset Description

YOLOv8 is trained on the COCO dataset [7] which has 80 classes that include person and sports ball. Therefore, to detect the primary classes mentioned, a custom dataset was needed to fine-tune the model. The classes included in this dataset are: Player, Ballhandler, Ball, Basket, and Made Basket.

The images used to generate this dataset were extracted from frames of the video: *Golden State Warriors vs Boston Celtics Full Game 3, 2022 NBA Finals* [8], which has a 1920x1080 resolution and 30 FPS. This implies that the dataset is very limited in terms of diversity. However, the development of a very generalized dataset that includes all the teams in all the arenas falls outside the scope of this paper.

After several iterations and dataset expansions, the dataset created [9] has 567 images, 6214 annotations, and a split between training and validation of 80-20. As there are usually 9 objects in each frame that belong to the player class, the dataset is imbalanced.

B. Stage 2: Player and Ballhandler Post-processing

Once the primary classes have been detected, the players and ballhandler objects have to be processed to identify the player and determine to which team belongs.

There are several approaches to player identification. One of these approaches is taking advantage of how humans traditionally differentiate teams and players, through jersey color and number classification [10] [11] [12]. Therefore, player identification requires jersey color analysis to determine the team of a certain player, and number recognition to determine the identity of that player.

1) Team Classification: Fisher's Discriminant.

A simple and efficient (low inference time) technique to classify a set of data is Fisher's Discriminant. The idea is to use the color information of the player image to determine the team. Each image, which represents a datapoint, has 3 features, the mean values of the 3 RGB channels of all the pixels in the image. Therefore, this 3-dimensional problem will be transformed into a 1-dimensional problem to determine a boundary between classes.

Fisher's Discriminant was developed by Ronald Fisher [13] and it is a generalization of the Linear Discriminant Analysis and equivalent to a special case of the Maximum A Posteriori (MAP) Bayesian decision rule, where the probability density functions (PDFs) are considered normal Gaussian distributions with equal covariance. The idea of this method is to find the feature projection that maximizes the distances between class PDFs to make a better decision.

To do so, the function $J(w)$ (separation of classes) has to be maximized:

$$J(w) = \frac{w^T(m_1 - m_2)(m_1 - m_2)^T w}{w^T(S_1 + S_2)w} \quad (1)$$

where m_i is the vector of the feature means for i, S_i is the *scatter* of features for class i, and w is the normal vector to the projection. The relationship is shown in equation (2).

$$S_i = N_i cov(x_i|w_i) \quad (2)$$

With all of this in mind, the w vector that maximizes the $J(w)$ function must be found, which will lead to equation (3).

$$w^* = S^{-1}(m_1 - m_2) \quad (3)$$

Once the 3-dimensional problem has been transformed into a 1-dimensional problem, a threshold must be selected to differentiate the two teams. As it was mentioned, a normal Gaussian distribution is going to be assumed for both PDFs and the probability of the true state of the classes is going to be considered the same ($P(w_o) = P(w_1)$). Therefore, the threshold selected will be the intersection of the PDFs.

To implement this method, a dataset was created by extracting 860 player images for each team in the game *Golden State Warriors vs Boston Celtics Full Game 3, 2022 NBA Finals* [8].

2) Player Identification: Number Recognition with EasyOCR.

As there can only be one player with a certain number in each team, once the team a player belongs to is determined and the jersey number is recognized, the player would be unambiguously identified.

Several methods can be used to recognize numbers on jerseys in different sports events [14] [15], but the approach followed in this paper is similar to the one presented by Ahmed Nady and Elsayed E. Hemaye on *Player Identification in Different Sports* [16].

From the previous stage of the BVA, a cropped image of a player is obtained. To detect and recognize the number on the player's jersey, a scene text detector followed by a scene text recognition model will be applied, using the software EasyOCR [17]. The structure of this software is based on Baek et al. work [18] on scene text detection (CRAFT) and Shi et al. work [19] on scene text recognition.

C. Stage 3: Software Integration and Statistics Generation

The last stage of the BVA is the algorithm designed to extract the game events (passes, attempted field goals, made field goals, and rebounds). This algorithm receives the information coming from the object detector and tracker and outputs the player and team statistics. The input received from YOLOv8 is the following for each object: bounding box, class, confidence of classification, and tracking ID. The output of the module is:

- Team and player statistics: passes, attempted field goals, made field goals, and rebounds.
- Primary class information for processed video representation.

An outline of this stage's architecture can be seen in Fig. 2.

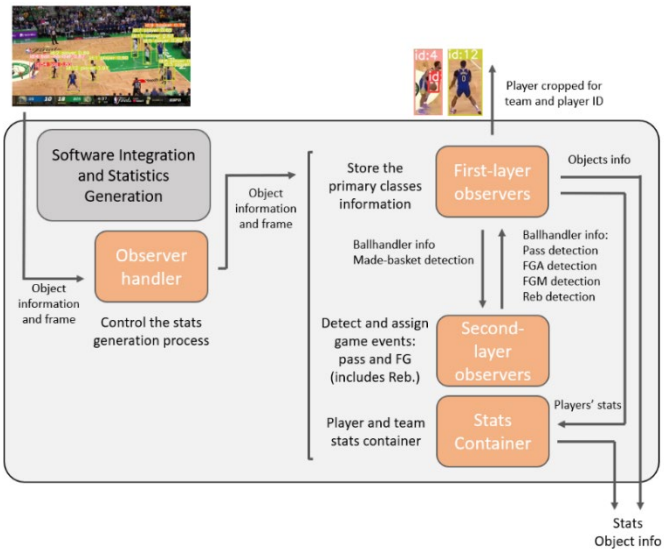

Fig. 2. Stage 3 architecture.

In this algorithm, the detection information and stats generation are processed by a set of classes that have been named observers. There are two types of observers, first-layer observers (primary classes containers) and second-layer observers (stats generators). The data generated by these two classes are reported to the stats container. All these classes are controlled by the observer handler.

1) First-Layer Observers

This type of class holds all the information related to the processed data coming from the object detector. There are 3 types of first-layer observers: *player_obs* (to enhance the ballhandler and team selection, this class adds a *counter* and a threshold so that only when a condition is met multiple times it is considered true), *basket_obs,* and *ball_obs.*

All the information held by these classes is used by the second-layer observers to detect game events and by the observer handler to export the processed video annotations.

2) Second-Layer Observers

This type of observer detects the mentioned game event: passes, FG attempts, FG made, and rebounds. There are 2 types of second-layer observers:

- *pass_obs*: with every frame, *pass_obs* receives the information of the current ball handler. Comparing this information with the previously detected ballhandler and their respective teams, this class assigns a passing event.
- *fg_obs*: this class generates the rest of the mentioned game events: FG attempts, FG made, and rebounds. With the method *check_posible_fg*, it checks if there is a bounding box intersection between the ball and the basket. If that is the case, an FGA is assigned and a timer is initiated to avoid recurrent detections. If by

the end of the FGA timer, a made basket has not been detected, a rebound is assigned. Otherwise, when a made basket is detected, another timer is initiated to avoid multiple FGM detections.

3) Statistics Container

The class *team* contains all the information related to players' information for each team and is used to report all the game information acquired from the video to the system.

4) Observer Handler

This last class is the one responsible for controlling all the stats generation and object detection post-processing for video annotation. This class has several methods to activate or deactivate first-layer objects, filter certain objects, and report first-layer objects for video annotation, but the most important function is the *upd_observers* which defines the algorithm to process the detections from the object detector and tracking algorithm.

IV. Results

A. Stage 1: Object Detection and Tracking. Results

The YOLOv8 model was trained on Google Colab. The dataset used to train the model was specifically designed for this work and it was thoroughly presented in the System Design section. In Table 1, an overview of the different parameters of the model training can be seen.

TABLE I. Model Parameters During Training.

Parameter	Value
Training samples	455
Validation samples	112
GPU model	Tesla T4 (16 Gb)
Number of layers	268
Number of Parameters	43610463
Epochs	100
Training time	1h and 2 mins
Inference time	21.2 ms
mAP50	0.835
mAP50-95	0.549

As we can see in Fig. 3, the model trained is almost perfect at detecting the basket and the players, which are sometimes misclassified as background or ballhandlers. This last misclassification is understandable given the similarity of these two classes. The ball and the made basket are classes that also have good results in terms of True Positives (TP), but the model sometimes confuses them with the background. Finally, the class with which the model struggles the most is the ballhandler class, commonly misclassified as a player, because the features of these two are very similar. Only by extending the dataset, better results could be obtained.

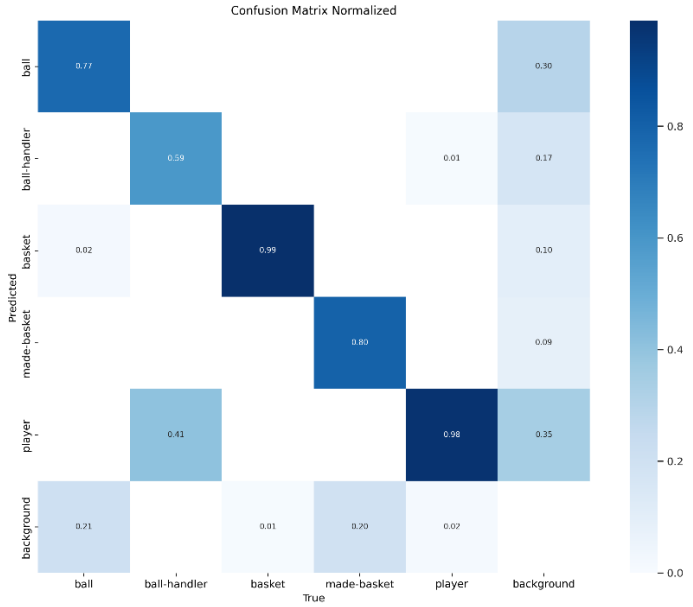

Fig. 3. Normalized confusion matrix.

In Fig. 4, the F1-confidence curve is displayed. As we can see, the model can detect all classes with good precision and recall (F1-score of 0.8 at 0.7 confidence) with some differences among the classes. For the player and the basket, the model achieves an F1-score of over 0.9 for 0.8 confidence. However, for the ball, the made basket, and the ballhandler, the F1-score oscillates between 0.65 and 0.75 until 0.8 confidence.

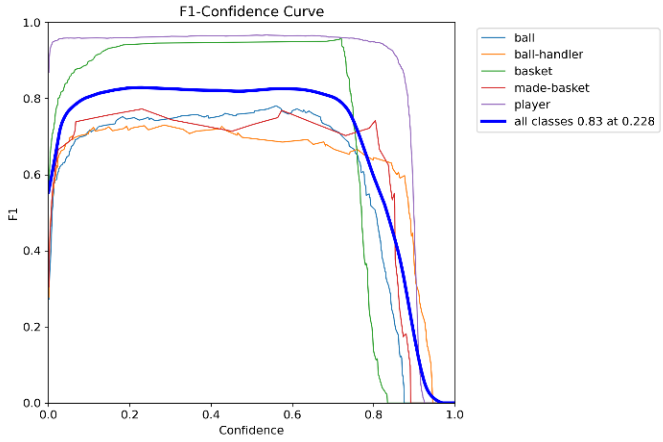

Fig. 4. F1-confidence curve.

This shows that the trained object detector has very good results detecting some of the classes and acceptable results detecting others.

B. Stage 2: Player and Ballhandler Post-processing. Results

In Fig. 5, the histogram of the 3 RGB channels for the 2 classes (teams) is displayed, which suggests that good results should be obtained when classifying the images based on this data.

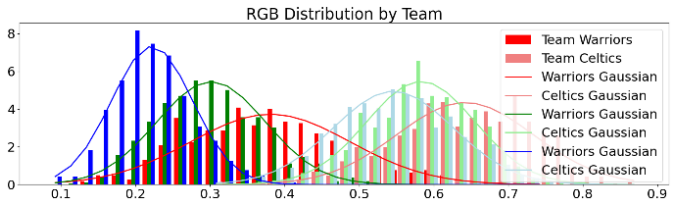

Fig. 5. RGB values distribution for both teams.

After projecting the data to the 1-dimensional space, the distribution is the one shown in Fig. 6.

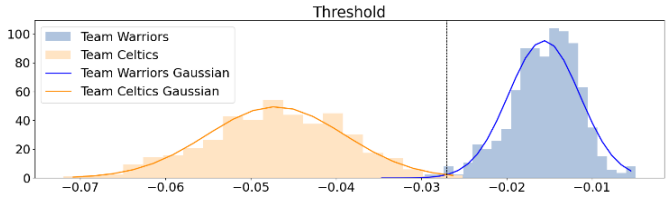

Fig. 6. 1-dimensional representation of the data and threshold.

With this information, the threshold is obtained as the intersection of the Gaussian distributions (-0.027).

The accuracy obtained from the training data is shown in Table 2. As we can see, the accuracy is excellent for both teams.

TABLE II. TEAM CLASSIFICATION RESULTS.

Parameter	Value
Training Warriors correct	98.95 %
Training Celtics correct	99.61 %
Training correct	99.28 %
Testing Warriors correct	95 %
Testing Celtics correct	100 %
Testing correct	97.5 %
Mean inference time	0.23 ms

A new dataset was created with 100 samples for each team for testing purposes. The accuracy obtained for this testing dataset is 97.5%, which is a very good result for a simple and quick algorithm. All in all, these results demonstrate a successful approach for team classification.

For player identification, the results obtained are presented in Table 3.

TABLE III. NUMBER RECOGNITION RESULTS.

Parameter	Value
Detections	45.1 %
Correct recognition among detection	78.26 %
Total correct recognitions	35.29 %
Mean confidence of correct recognitions	93.15 %
Mean confidence of incorrect recognitions	42.11 %

As we can see, the system has a relatively low accuracy of 35% in recognizing the numbers. However, the system detects almost half of the numbers in the dataset, and among those detections, the recognition accuracy is near 80%. This means that this 2-stage model struggles in the detection part and has good accuracy in identifying the number detected. This fact is acceptable as the BVA is working with video and not with independent frames. Therefore, a medium recall but low false positive is an acceptable performance for the number recognition model, as if a number is not detected in a frame, it could be detected and correctly identified in the subsequent.

Moreover, when the number is correctly recognized, the confidence is higher than 90%, and when it is a false positive, the confidence is near 40%. With this information, a threshold can be applied to limit the number of incorrect recognitions. All in all, this model is suitable for number recognition.

C. Basketball Video Analysis System. Results

To check the correct operation of the system, it was tested on 40 clips with a total of 103 events. The results can be seen in Table 4.

TABLE IV. EVENT DETECTION RESULTS.

Parameter	Value
Passes detected	76.57 %
FGA detected	93.9 %
FGM detected	100 %
Rebounds detected	50 %
Event assigned to correct team	50.59 %
Event assigned to incorrect player	27.18 %
Inference time (without YOLO)	~60 ms

As we can see, the system obtains good results when detecting the passes, and excellent ones in terms of FGA and FGM detections, but not in rebound detection. First, the passes are detected less than the FGs because of the object detector struggling to detect the ballhandler, affecting the pass detection which depends solely on it. Secondly, the reason behind the low detection of the rebounds is also the ballhandler, as in these situations the scene is usually very crowded, and it is very difficult to distinguish the ballhandler.

Furthermore, something relevant to test is the number of times the system assigns an event to the correct team and the wrong player. Here, again, the ballhandler detection plays an important role because in certain situations, especially when a player has shot the ball, the system is not consistent with the ballhandler selection which leads to an incorrect assignment of the event.

Finally, the inference time was calculated for the BVA, obtaining 12.5 FPS when using Google Colab (Tesla T4). Therefore, with the right hardware, this system could be used in real-time applications.

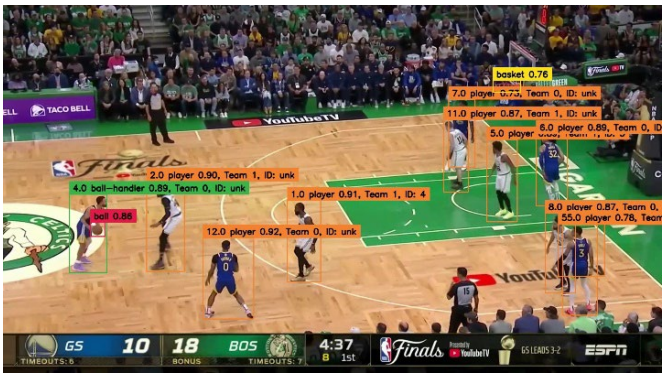

Fig. 7. Frame from an output processed video.

V. CONCLUSIONS

This work presents a solution for a system able to automatically acquire relevant basketball game data from a broadcast video. The final goal of the paper is to generate statistics that are commonly used to analyze the performance of players and teams. Among these statistics, the ones that are considered for this paper are passes, field goal attempts, field goals made, and rebounds.

To achieve this objective, a 3-stage system is proposed. The first stage consists of an object detector (YOLOv8) combined with a tracking algorithm (BoT-SORT) that can detect and track the main elements present in a basketball game video: players, ballhandler, ball, basket, and made basket, also referred to as primary classes. To correctly assign the generated statistics to the correct team and player, player identification is needed. The second stage of this work proposes a player identification process based on two phases, team classification and number recognition. For the former, Fisher's Discriminant is used to classify the team based on the RGB pixel values of the image. For the latter, a 2-stage OCR architecture is used (EasyOCR). Finally, the last stage of the BVA system, with the information from the previous stages, generates the mentioned statistics and assigns them to the corresponding player and team.

The results obtained from these three stages suggest that the architecture selected to achieve the goals defined is a promising approach. The system consistently detects the game events defined, struggling with some events like rebounding and with player and team assignation. This last issue is mostly related to the first and second stages of the system, but with certain improvements (dataset extension and algorithm improvement), better results would be expected.

All in all, the system designed in this work could be used in several scenarios resulting in very helpful in most situations, like box scoring aid in NBA games, statistics annotation for amateur basketball games, or data acquisition for post-game analysis.

REFERENCES

[1] Sports analytics: How different sports use data analytics. [Online]. Available: https://www.datacamp.com/blog/sports-analytics-how-different-sports-use-data-analysis.

[2] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[3] MOT challenge. [Online]. Available: https://motchallenge.net/

[4] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," 2017.

[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.

[6] N. Aharon, R. Orfaig, and B.-Z. Bobrovsky, "Bot-sort: Robust associations multi-pedestrian tracking," 2022.

[7] COCO - common objects in context. [Online]. Available: https://cocodataset.org/#home

[8] NBA Highlights, "Golden state warriors vs boston celtics full game 3 highlights | 2022 NBA finals." [Online]. Available: https://www.youtube.com/watch?v=04lMj66I5x8

[9] 597, "bva dataset," https://universe.roboflow.com/597/bva, jul 2023, visited on 2023-07-25. [Online]. Available: https://universe.roboflow.com/597/bva

[10] T. Guo, K. Tao, Q. Hu, and Y. Shen, "Detection of ice hockey players and teams via a two-phase cascaded cnn model," *IEEE Access*, vol. 8, pp. 195 062–195 073, 2020.

[11] Y. Yoon, H. Hwang, Y. Choi, M. Joo, H. Oh, I. Park, K.-H. Lee, and J.-H. Hwang, "Analyzing basketball movements and pass relationships using real-time object tracking techniques based on deep learning," IEEE Access, vol. 7, pp. 56 564–56 576, 2019.

[12] R. Alhejaily, R. Alhejaily, M. Almdahrsh, S. Alessa, and S. Albelwi, "Automatic team assignment and jersey number recognition in football videos," *INTELLIGENT AUTOMATION AND SOFT COMPUTING*, vol. 36, no. 3, pp. 2669–2684, 2023.

[13] P. E. Hart, D. G. Stork, and R. O. Duda, Pattern classification. Wiley Hoboken, 2000.

[14] H. Liu and B. Bhanu, "Jede: Universal jersey number detector for sports," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 11, pp. 7894–7909, 2022.

[15] D. Bhargavi, E. P. Coyotl, and S. Gholami, "Knock, knock. who's there?-identifying football player jersey numbers with synthetic data," *arXiv preprintar Xiv:2203.00734*, 2022.

[16] A. Nady and E. E. Hemayed, "Player identification in different sports." in *VISIGRAPP (5: VISAPP)*, 2021, pp. 653–660.

[17] EasyOCR," original-date: 2020-03-14T11:46:39Z. [Online]. Available: https://github.com/JaidedAI/EasyOCR

[18] J. Baek, G. Kim, J. Lee, S. Park, D. Han, S. Yun, S. J. Oh, and H. Lee, "What is wrong with scene text recognition model comparisons? dataset and model analysis," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 4715–4723.

[19] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," 2015.