Optimizing Basketball Shot Trajectory using Image Segmentation Techniques for Training Feedback

Vasisht Kartik, Ross Greer

Abstract—In basketball, the efficiency of shot-making is crucial for success, especially in high-stakes games. Traditional self-training methods face significant challenges due to the absence of training equipment and personalized coaching. This paper presents an approach using computer vision and deep learning algorithms to provide feedback to players towards optimizing basketball shot trajectories, offering a solution for self-training. We employ image segmentation to accurately track the basketball and analyze shooting videos, enabling the extraction of critical parameters such as the release angle and shot trajectory. Our methodology integrates a Faster R-CNN model for object detection and introduces two novel parabolic curve fitting techniques: Bounce-Around and Sliding Window Sampling Consensus (SWISAC). These techniques allow for precise trajectory analysis and on-or-off-course predictions, despite occlusions by the net. Experimental results demonstrate the efficacy of our approach in providing actionable feedback for improving shooting accuracy. This research lays the groundwork for future advancements in automated sports analytics, enhancing and democratizing the training and performance feedback of basketball players.

Index Terms—intelligent feedback systems, sports analytics, object tracking, human-computer interaction, basketball

I. INTRODUCTION

The game of basketball is a highly competitive sport with a massive worldwide fan base. This sport requires various skills such as dribbling, passing, and rebounding, culminating in successful basketball scoring. The outcome of most high-level games is determined by the team that shoots the basketball more efficiently. While basketball requires a great level of teamwork and coordination, individual shooting skills are extremely crucial to achieving success. Insufficient shooting abilities can expose many basketball teams' weaknesses and can ultimately reduce chances for team success. Therefore, shooting efficiently in basketball plays a key role in securing wins. Intense and repeated self-training is needed to improve basketball shooting abilities. However, self-training poses a challenge in that it is not easy to analyze one's own basketball shooting method and identify adjustments to be made. It is also not feasible for every basketball player to have an individual coach [1]. These obstacles can be effectively countered by having technology tools that provide recommendations for improving basketball scoring techniques. These tools have computer vision and image processing at their core. Computer vision algorithms and deep learning have been utilized to develop image segmentation techniques, addressing Fig. 1. We used an R-CNN pre-trained model to detect the ball right before it enters the basket. We utilize Mediapipe to detect the pose and hand features to ensure bodily features are not considered as the basketball. real-world problems like pose detection [2]–[4] and object detection [5]. Despite their numerous uses, deep learning algorithms have some disadvantages. Deep learning, for instance, requires higher computing power and extensive datasets to complete even simple image segmentation tasks. Image segmentation is a key tool for our research purposes since it enables precise identification of the region containing the target object and facilitates an accurate estimation of the target object's center, a common complexity of object representation [6]. Image segmentation applied to a sequence of individual video frames plays a pivotal role in analyzing and tracking the trajectory of objects in dynamic environments [7], [8]. By isolating only the target object, image segmentation techniques can minimize unnecessary elements and better depict an object's trajectory. In this paper, we determine the trajectory of a basketball by analyzing the shooting videos using computer vision, image segmentation, and data analysis techniques. Using this information, we create an algorithm to estimate whether a shot is a on-course or off-course, for the purpose of providing feedback on the shot angle. However, we find that state-of-the-art object detection methods are only able to detect the basketball up until the point where the ball becomes occluded by the net. While we can still determine a parabola of the projected motion of the ball, due to the unpredictability of interactions between the ball and the rim, we make the case for future research in the detection of the basketball through Fig. 2. State-of-the-art algorithms were unable to detect the basketball while in the net. Having this advanced capability would assist in estimating whether the shot was a make or a miss net occlusion.

Fig. 1. We used an R-CNN pre-trained model to detect the ball right before it enters the basket. We utilize Mediapipe to detect the pose and hand features to ensure bodily features are not considered as the basketball.

Fig. 2. State-of-the-art algorithms were unable to detect the basketball while in the net. Having this advanced capability would assist in estimating whether the shot was a make or a miss.

Fig. 3. The orange bounding box is used to detect the rim. In this frame, the algorithm detects both the rim as well as a clock to be the rim, a common outlier we encountered.

II. RELATED RESEARCH

Computer vision and deep learning algorithms have been utilized to develop effective segmentation techniques [9], addressing real-world problems like pose detection [10] and object detection [11]. Similar techniques have also been applied for sports analysis, with relevant examples for basketball feedback provided in this section. In [12], a 3x3 red median filter is used along with binarization and area expansion to identify the red ball and shirt that the player is wearing. The centroids of the player's posture and the basketball are identified by detecting the basketball and the player's t-shirt. Using the centroids and the given height of the person, the optimal shooting angle is predicted. The basketball trajectory is estimated using starting angle and refined using a Kalman filter. In this paper, we also determine an optimum shooting angle as in [12] using an alternate predicted on-or-off course algorithm at the rim to enable the method to work for any player's height while accounting for key factors such as the rim or the backboard affecting the ball's trajectory. [13] studies the postural attributes such as leg bending, stomping, ball lifting, elbow lifting, and arm stretching upward as in [12] and then identifies correlations between these postural attributes and the shot result (make or miss). This approach is currently limited to free throws and does not provide a path to extend it for other shooting poses on the basketball court. Instead, the focus of our approach is recognition of the basketball and the rim and works on extrapolating the basketball trajectory. Our approach is shown to also recognize the basketball and rim when shooting from different court locations and different angles of recording videos but narrows down to summarize one-or-off course algorithm only for free throws. [14] proposes a dual-core extreme learning machine (ELM) combined with deep learning for motion recognition, which is evaluated on large-scale and real-world datasets. It uses a combination of deep learning methods to recognize and analyze basketball shooting angles. We address a similar task Fig. 3. The orange bounding box is used to detect the rim. In this frame, the algorithm detects both the rim as well as a clock to be the rim, a common outlier we encountered. in our research, and further include a mechanism for feedback and self-coaching, particularly leveraging pre-trained models to recognize the rim to assess shot trajectory targets.

III. METHODOLOGIES

In this paper, we first determine the trajectory of the basketball by analyzing the shot-making videos using computer vision and image segmentation techniques. We developed two parabolic curve fitting mechanisms, one consisting of forward and backward iterations on the ball's trajectory from its apex point and another consisting of a sliding window through the time series of basketball detection. Both methods are used to identify two critical aspects at the endpoints of the arc of a basketball shot, which are the release angle and the point before the ball hits the basket. This information is used in providing on- or off-course predictions and calculating release angles. The algorithm's prediction is able to run independently for each person, so that it adjusts to the appropriate angle for any player's height.

A. Object Detection - To detect the basketball and rim in a given frame, we used pre-trained Faster R-CNN model [15], trained on the Common Objects in Context (COCO) dataset [16]. Within each frame, we obtained the ball and rim class labels, bounding boxes for the two classes, and their confidence scores. We configured the ball and rim detection filter for a confidence threshold of 0.5. There were false positives when identifying both the ball as well as the rim. The subsequent sections describes the algorithm we developed to filter out such false ball or rim detection. In each frame, we store the coordinates of the bounding box around the ball and the rim as well as the ball's center position in a data frame for further processing.

B. Filtering Out False Positives

Initially, the test videos were taken in an outdoor setting, where a player's head was the only object that was found to create a false positive. There were no false positives for rim detection outdoors. We observe that once the basketball has left the player's hand in a shot-making process, the basketball is always higher in the subsequent frames. This fact was used to develop the filter to remove false positives. In every subsequent frame, whenever more than one ball is detected, we save the ball position coordinate that was closer to the ball position coordinate from the previous frame while considering the outlier ball detection as extraneous coordinates. When conducting test videos in an indoor setting, there were false positives other than the player's head for both ball and rim detection. To ensure that false positives were filtered in both indoor and outdoor settings, we refined the algorithm to retain the coordinates for both the ball and the rim which was closer to the previous frame's ball and rim position coordinates. It is imperative that the ball and the rim are detected accurately in the first frame to prevent future outliers, and the confidence threshold of 0.5 worked sufficiently for our task.

C. Parabola Curve Fitting

1) Bounce-Around Method: When a basketball is released from a player's hand, it travels under the influence of gravity, following a parabolic path. Under these assumptions, the only factors that alter the ball's trajectory are interactions with the backboard, rim, or net. To determine the trajectory of a basketball, we can fit a curve to the ball's path using its position coordinates. We first identified the apex point of the ball's trajectory as the highest y-coordinate of the ball position coordinates, and curve fit a parabola using one point on either side of the apex (3 points total).

The, a, b, and c coefficients of the quadratic fit curve were extracted and stored.

$$f(x) = ax^2 + bx + c \qquad predicted\ equation\ of\ parabola$$

This parabola does not depict the full trajectory of the ball. To replicate the full path of the ball, we found two endpoints, the start endpoint, which signifies the moment the ball leaves the player's hand and the final endpoint, which is when the ball is about to come into contact with the rim or backboard. To do this, we took one point on either side of the apex point of the curve fit parabola and continued to iteratively do this until the proportion of variance (R2) values on both the left and right side became less than 0.99. These two points on the left and right side were considered the endpoints of the ball's trajectory in the air. The start endpoint was considered as the point where the player releases the ball from the hand and the final endpoint as when it is about to interact with the hoop.

$$R^2 = 1 - \frac{\text{sum squared regression (SSR)}}{\text{total sum of squares (SST)}} = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \overline{y})^2}$$ (1) 2∑(yi − y)2

(1)
We then curve fit a final parabola using all the ball coordinate points between these two endpoints and saved the coefficients. This method allows us to determine an accurate representation of the ball's path but does not account for any noise in the video or detections.
2) Sliding Window Sampling Consensus (SWISAC): The second method employed to achieve an optimal parabola fitting utilized a variant of the RANSAC (Random Sample Consensus) regression model with a sliding window instead of random selection. Our approach, which we name SWISAC, involves using a sliding window of five ball position coordinates to fit a parabola. Starting with the first ball position, the method selects the initial five points to curve fit a parabola. The window then moves to the next ball position and repeats the process for the subsequent sets of coordinates until the end of the time series of detections is reached. For every set of five ball position coordinates, we calculated the R2 value, as shown in the graph in Figure 5. We then identify the longest set of contiguous windows that had R2 values consistently over 0.97 while still allowing for a certain number of windows to have R2 values lower than the 0.97 threshold (tolerance value = 3). This segment represents the trajectory of the ball while it is in the air. To fit the resultant parabola, we took the sliding window's maximum R2 value parabola and used this parameterization, storing the coefficients a, b, and c of this resultant parabola.

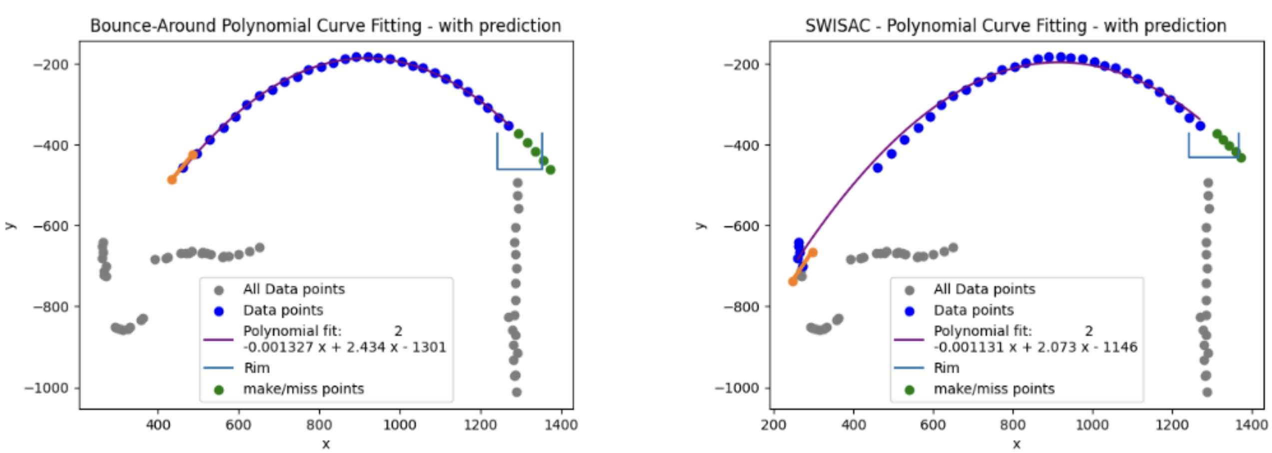

Fig. 4. Curve fit parabola through iterations of points on either side of the apex point (bounce-around, left) and using the sliding window's maximum R^2 value parabola to curve fit on the ball trajectory points (SWISAC, right). The blue box on the right side represents the rim and the gray points signify the ball position coordinates that were not used for the final curve fitting. The ball is never detected in the rim's vicinity due to the occlusion of the net. The orange line in the beginning is the shooting angle determined by analyzing the first two points from the beginning of the shot. The green points at the end of the parabola is the ball's projected path based on the equation of the parabola for the Bounce-Around and SWISAC method.

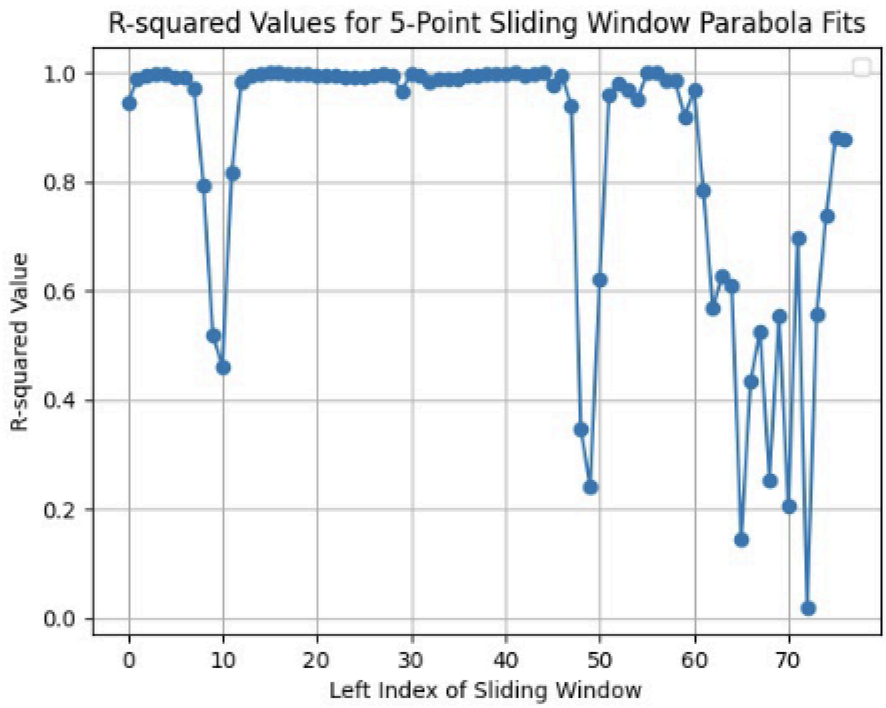

Fig. 5. The x-axis represents the starting index of each set of five ball position coordinates. There is a long segment of (R^2) values holding values ¿ 0.97. This segment is considered to be the real ball trajectory from the time it leaves the player's hand to when it interacts with the hoop.

D. On-or-Off-Course Algorithm

Make or miss algorithms rely on ball detection while it is in the hoop. As a substitute, we extend the curve fit parabola and project 5 different points and determine the ball's trajectory was on-course or off-course in entering the rim for algorithmic purposes.

1) Define n (n = 5, tunable parameter) y coordinates points between the top and bottom of the rim
2) Using these n y-coordinates, predict the 5 x-coordinates from the parabola curve fit equation and store them
3) If at least m (1, tunable) points fall within the rim rectangle boundary, the shot is considered a make, otherwise a miss

E. Angle Detection

Utilizing the ball trajectory parabola's coefficients, we defined a derivative function to calculate the angle at the first ball position coordinate of the curve fit parabola. The derivative for any given parabola is in the form

$$\frac{dy}{dx} = 2ax + b$$

and converted into radians and degrees. After calculating the angle in degrees, we drew a line with length 50 through the first ball position coordinate to visualize the angle and the position it was calculated with. Figure 4 shows the final plot with real, predicted on-or-off course basketball trajectory, and angle of release. For each parabola graph shown in the previous sections, the orange line represents a line that shows the angle for that particular video.

IV. EXPERIMENTAL EVALUATION

Data was collected from various shooting locations, and recorded at different angles for the same shot location. Outdoor basketball shots did not face significant issues, since there were fewer external objects. Indoor shots faced challenges with extraneous items on the wall (e.g., clock, fire alarm) being misconstrued as the basketball or rim, and passersby i interfering with the shot. To recognize our algorithm's effectiveness, we took the ground truth of the shot (make vs miss) and the predicted output of the ball (on-course or off-course) from free-throw line shots and stored the data. Our model was able to predict correctly 66 percent of the real-time free-throw line shots using Bounce-Around parabolic curve fitting and our estimated make-or-miss algorithm. For the SWISAC parabolic curve fitting, our make-or-miss algorithm predicted correctly 68 percent of real-time free-throw line shots.

1) Video Specifications: Every video that our model evaluated was shot on an iPhone at 60 fps which resulted in our model processing 60 frames per second. We collected 58 total videos of basketball shots with 34 of the shots being from the free-throw line. 25 of the shots from the free-throw line were indoors. An orange indoor basketball was used indoors while a green ball was used outdoors to eliminate similarity. Every video we analyzed had single-player shots rather than multi-player shots to ensure that our model only tracked one basketball.

2) Processing: We processed all frames without skipping any to ensure comprehensive analysis. We tested both the Bounce-Around method and the SWISAC method for every free-throw video taken. For every output frame, we stored the output for the ball, rim, hand, and rim's coordinates along with the final trajectory plots in an output file. In the output file, we included predictions (On-Course or Off-Course) as well as the unique shooting angles for different shots. The plots of the shooting angles are intended to provide visualized recommendations for altering shooting angles appropriately. We provide angles and plots for both the Bounce-Around method and the SWISAC method. We processed 30 free-throw videos for the Bounce-Around method (Table 1) and 22 free-throw videos using the SWISAC method (Table 2).

TABLE I
BOUNCE-AROUND METHOD PREDICTED ON-OR-OFF COURSE VS GROUND TRUTH MAKE OR MISS

	Predicted On-Course	Predicted Off-Course
Ground Truth Make	16	3
Ground Truth Miss	7	4

TABLE II
SWISAC METHOD PREDICTED ON-OR-OFF COURSE VS GROUND TRUTH MAKE OR MISS

	Predicted On-Course	Predicted Off-CoursePr
Ground Truth Make	12	4
Ground Truth Miss	3	3

A. Experiment Results and Feedback Mechanism All the free-throw line basketball shots are stored with important details about the shot information. The video name, shooting location (indoor or outdoors), video angle (the spot on the court), court location (right or left), On-or-Off course prediction, and shooting angle are stored. We take the median angle from our algorithm's predicted on-course shots and store it. If the user shoots with an angle higher than this, a lower shooting arc would be suggested to the user. Vice versa, if the user shot with a lower angle than the median angle, a higher arc would be suggested. For example, for free-throw line shots using the Bounce-Around method, the optimum angle our algorithm suggests for the player shown in Figure 1 52.3°. For free-throw line shots using the SWISAC method, the optimum angle suggested is 55.6°. The consistency of angles between both methods is an encouraging feature of these approaches.

V. CONCLUDING REMARKS AND FUTURE RESEARCH

This paper presents an approach utilizing computer vision and deep learning algorithms to provide actionable feedback for optimizing basketball shot trajectories, offering a robust solution for self-training. By employing image segmentation, we accurately track the basketball and analyze shooting videos, enabling the extraction of critical parameters such as release angle and shot trajectory. Our methodology integrates a Faster R-CNN model for object detection and introduces two novel parabolic curve fitting techniques: Bounce-Around and Sliding Window Sampling Consensus (SWISAC). These techniques allow for precise trajectory analysis and on-or-off-course predictions, even in cases of occlusion by the net. Experimental results demonstrate the efficacy of our approach in improving shooting accuracy. We also lay the groundwork for a feedback mechanism that provides automated suggestions to players based on the median shooting angle vs the angle they shot at. Towards future research, our make-or-miss algorithm relies on detecting the basketball while it is in the net, which is a current occlusion challenge for object detectors. We hope to further research that enables state-of-the-art object detection mechanisms to detect the ball while in the net for our make or miss algorithm to return feedback not only on trajectory but also on shot success. Using active learning techniques [17]–[21] may assist in identifying challenging examples for annotation and learning, allowing for the training of detection models which are robust to occlusion and providing a consistent and useful analysis of a player’s shot for self-coaching.

REFERENCES

[1] B. Abdoli, J. Hardy, J. F. Riyahi, and A. Farsi, “A closer look at how self-talk influences skilled basketball performance,” The Sport Psychologist, vol. 32, no. 1, pp. 9–15, 2018.
[2] Z. Fan, Y. Zhu, Y. He, Q. Sun, H. Liu, and J. He, “Deep learning on monocular object pose detection and tracking: A comprehensive overview,” ACM Computing Surveys, vol. 55, no. 4, pp. 1–40, 2022.
[3] C. Zheng, S. Zhu, M. Mendieta, T. Yang, C. Chen, and Z. Ding, “3d human pose estimation with spatial and temporal transformers,” in Proceedings of the IEEE/CVF international conference on computer vision, 2021, pp. 11 656–11 665.
[4] R. Greer and M. Trivedi, “Ensemble learning for fusion of multiview vision with occlusion and missing information: Framework and evaluations with real-world data and applications in driver hand activity recognition,” arXiv preprint arXiv:2301.12592, 2023.
[5] A. Ghita, B. Antoniussen, W. Zimmer, R. Greer, C. Creß, A. Møgelmose, M. Trivedi, and A. C. Knoll, “Activeanno3d-an active learning framework for multi-modal 3d object detection,” in 35th IEEE Intelligent Vehicles Symposium (IV) 2024, 2024.
[6] R. Greer, A. Gopalkrishnan, M. Keskar, and M. M. Trivedi, “Patterns of vehicle lights: Addressing complexities of camera-based vehicle light datasets and metrics,” Pattern Recognition Letters, vol. 178, pp. 209–215, 2024.
[7] P. Sun, J. Cao, Y. Jiang, Z. Yuan, S. Bai, K. Kitani, and P. Luo, “Dancetrack: Multi-object tracking in uniform appearance and diverse motion,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 20 993–21 002.
[8] R. Greer, S. Desai, L. Rakla, A. Gopalkrishnan, A. Alofi, and M. Trivedi, “Pedestrian behavior maps for safety advisories: Champ framework and real-world data analysis,” in 2023 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2023, pp. 1–8.
[9] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz, and D. Terzopoulos, “Image segmentation using deep learning: A survey,” IEEE transactions on pattern analysis and machine intelligence, vol. 44, no. 7, pp. 3523–3542, 2021.

[10] Z. Ivankovic, M. Rackovic, and M. Ivkovic, “Automatic player position detection in basketball games,” Multimedia tools and applications, vol. 72, pp. 2741–2767, 2014.
[11] M. Buri´c, M. Pobar, and M. Iva˘si´c-Kos, “Object detection in sports videos,” in 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO). IEEE, 2018, pp. 1034–1039.
[12] Y. Egi, “Basketball self training shooting posture recognition and trajectory estimation using computer vision and kalman filter,” Journal of Electrical Engineering, vol. 73, no. 1, pp. 19–27, 2022.
[13] B. Zhao, “Deep learning-based basketball free throw attitude analysis and hit probability prediction system research,” Applied Mathematics and Nonlinear Sciences, vol. 9, no. 1.
[14] Y. Wang, M. Sun, and L. Liu, “Basketball shooting angle calculation and analysis by deeply-learned vision model,” Future Generation Computer Systems, vol. 125, pp. 949–953, 2021.
[15] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” Advances in neural information processing systems, vol. 28, 2015.
[16] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Doll´ar, and C. L. Zitnick, “Microsoft coco: Common objects in context,” in Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13. Springer, 2014, pp. 740–755.
[17] R. Greer, B. Antoniussen, M. V. Andersen, A. Møgelmose, and M. M. Trivedi, “The why, when, and how to use active learning in large-data-driven 3d object detection for safe autonomous driving: An empirical exploration,” arXiv preprint arXiv:2401.16634, 2024.
[18] S. E. Matos Flores, “Semi-automatic basketball jump shot annotation using multi-view activity recognition and deep learning,” in International Conference on Human-Computer Interaction. Springer, 2023, pp. 483–490.
[19] R. Greer and M. Trivedi, “Perception without vision for trajectory prediction: Ego vehicle dynamics as scene representation for efficient active learning in autonomous driving,” arXiv preprint arXiv:2405.09049, 2024.
[20] R. Greer, B. Antoniussen, A. Møgelmose, and M. M. Trivedi, “Language-driven active learning for diverse open-set 3d object detection,” in Vision and Language for Autonomous Driving and Robotics Workshop, CVPR, 2024.
[21] S. Ai, J. Na, V. De Silva, and M. Caine, “A novel methodology for automating spatio-temporal data classification in basketball using active learning,” in 2021 IEEE 2nd International Conference on Pattern Recognition and Machine Learning (PRML). IEEE, 2021, pp. 39–45.