Trajectory Detection And Boundary Estimation For Badminton Shuttlecock Using 3D Position Estimation

Peter Emeke Nsaka
Stanford University
nsaka@stanford.edu

Bryan Olisaemeka Mbanefo
Stanford University
bmbanefo@stanford.edu

Yaron Katriel Sternberg
Stanford University
yaronks@stanford.edu

Abstract - Badminton is a popular sport around the world and similar to other sports, a line judgment call can often decide games. Additionally, object-tracking for sports events is an ever-growing field of interest in computer vision. In this project, we set off with a goal to build an Instant Review System (IRS), commonly referred to as a Challenge System, for badminton, which allows players to challenge line judge decisions and promotes fairness in the game. We have been focusing on tracking a shuttlecock throughout the duration of a badminton game with the goal of determining if the shuttlecock is inside or outside of the court boundary. The unique flight characteristics and size of a shuttlecock also pose additional challenges. Factors like lighting, air conditioning, and moving mats can also significantly impact the decision of whether the shuttlecock lands in or out. In this paper, we discuss the issues we faced while developing the proposed solution. We describe the evolution of the system's architecture as well as provide a detailed analysis of the system's performance and compare it to factual decisions.

1. Introduction

Badminton is widely considered to be the fastest sport in the world, with the shuttlecock reaching speeds of over 400 km/h. In fact, a laboratory measurement in 2013 recorded a world record speed of 493 km/h for the shuttlecock. As a result, accurately determining whether a shuttlecock is in or out of bounds during a badminton game, given its incredibly fast speed can be challenging for an umpire due to human perception limitations.

To help umpires overcome these limitations, Instant Review System (IRS), commonly referred to as a Challenge System has been built. However, it is noted that existing systems are not always 100% accurate due to errors caused by occlusions, and improvements are still required to detect fast-moving small objects like shuttlecocks.

In this paper, we attempt to create a review system with cameras placed at various angles in the court to detect the exact location where the shuttlecock hits the ground by using 3D position estimation, leveraging techniques from Epipolar geometry. We also discuss approaches used in established research topics such as object detection, segmentation, and tracking, and highlight the use of algorithms and deep-learning techniques.

In the Technical Approach section, we detail our strategy for creating a shuttlecock tracking system that is versatile enough to function in real-world situations. The section also outlines the obstacles that were encountered during development and how they were successfully resolved. The Evaluation section examines the outcomes and efficacy of the system, while also contemplating potential approaches for addressing any remaining challenges that were could not be contained in the scope of the project due to time limitations.

2. Background and Related Work

Several research studies have focused on developing algorithms and methodologies to track shuttlecocks in videos, both in real-time and post-processing.

One approach proposed by Chen et al. [7] is the Fast-Tracking based on the Object Center method, which utilizes the Adaboost algorithm and heterogeneous cues to enhance the tracking performance of a 3D shuttlecock tracking system for a robot.

Another method proposed by Swalaganata et al. [4] combines the Camshift method and Kalman filter to improve tracking accuracy. The Camshift method is advantageous in diverse object color conditions, while the Kalman filter approach is useful for predicting object motion in the next frame based on the previous frame.

Shishido et al. [6] proposed a method for estimating the 3D position of a shuttlecock from videos taken by multi-view cameras. It observes the fast-moving target object as a curved line, and the shape of the silhouette technique is combined with Kalman filters to estimate the 3D position of a moving shuttlecock. The estimated 3D trajectory is

highly accurate, even when the shuttle moves exceptionally fast and erratically.

Vrajesh1 et al. [5] explored the use of deep learning models for detecting shuttlecocks. The first part of the work deals with the detection of the shuttlecock using deep learning techniques and the rest of the paper deals with the prediction of the shuttlecock using Neural Networks. The obtained prediction is around 80 percent accurate.

The primary challenge for scientists and developers is improving the speed and accuracy of ball detection, especially for fast-moving small objects like shuttlecocks. Detecting shuttlecocks still poses a significant challenge compared to detecting larger objects, with relatively low performance and requiring substantial hardware resources.

3. Technical Approach

The task is to precisely identify the landing location of a small and irregularly shaped shuttlecock that travels at different speeds throughout a badminton match.

To solve this problem, we broke it down into 5 subtasks which include:

Task 1 - Data Gathering: We start off by curating data sets necessary to help us build the aforementioned system. In some cases, the data was readily available online and in some cases, we need to generate the data such as a video of a badminton game, and camera calibration to get intrinsic camera properties. More on this is covered in Subsections 3.1 and 3.2

Task 2 - Model Training: Second, we trained and evaluated the performance of the YoloV5 model to detect the initial position of the shuttlecock. By utilizing YoloV5, we were able to estimate the starting position of the shuttlecock in any image frame. More on this is covered in Subsections 3.1 and 3.3.2

Task 3 - Court Boundary Detection: We extracted the court boundaries from the image frames. Combining the information with the camera parameters and the court boundary, we can estimate the 3D dimensions of the court. More on this is covered in Subsection 3.4

Task 4 - Shuttlecock Terminal Position Estimation: Next we apply a Kalman filter to the shuttlecock's position to predict its terminal position. The initial position of the shuttlecock used by the Kalman filter is detected by the YoloV5 model prediction done in Task 2 More on this is covered in Subsection 3.5

Task 5 - Evaluation: Finally, by leveraging results from Task 3 and Task 4 we determine if the terminal position of the shuttlecock as provided by the Kalman filter is within the bounds of the court.[6]. More on this is covered in subsection 3.6

Below is a diagram summarizing the details of the system in relation to the tasks needed to arrive at the expected outcome.

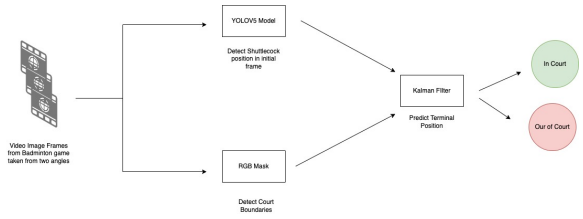

Figure 1. Sequence diagram summarizing technical approach

3.1. Data Generation

Two distinct sets of data were utilized. One for the purpose of training and testing a model for the automated detection of a shuttlecock in an image, and the other for evaluating the efficacy of trajectory detection and boundary estimation of the shuttlecock.

The first set of data consisted of a labeled dataset of shuttlecock images, containing 4500 images [3] of shuttlecocks in various positions and orientations. This dataset was used to train and test the model, with a split of approximately 60:40 between training and testing data.

On the other hand, the second dataset needed for trajectory detection and boundary estimation was challenging to obtain. To estimate if a shuttlecock is within or outside the boundaries of a court using Epipolar geometry we need the intrinsic properties of the camera used in video generation as well as at least two camera views of the game. Unfortunately, it was impossible to locate readily available examples of such datasets online or from related research projects. We attempted to use the first dataset utilized in training the shuttlecock detection model, however, it lacked camera parameters and multiple camera views of the same timeframe in the game.

Additional effort was made to create our own dataset. A badminton game was recorded in a badminton club at a local sports center using two Apple smartphones - an Apple iPhone X and an Apple iPhone 13 Pro Max - positioned at an elevated location of approximately 3.8 meters above the court. Videos were captured from opposite ends of the court, providing a good perspective. The phones were set to record video at a resolution of 3840×2160 pixels and a frame rate of 60 frames per second. The game was played under normal lighting conditions, with four experienced badminton players playing a doubles game. However, the dataset generated was unusable due to the presence of multiple courts in the video, introducing a high level of noise and making it nearly impossible to detect the court boundary.

For our second dataset, we ultimately decided to use a video of a badminton game captured from a single camera view found on Roboflow for trajectory detection and boundary estimation of the shuttle cock.

Figure 2. Image Frame From Dataset Taken using Camera 1

Figure 3. Image Frame From Dataset Taken using Camera 2

3.2. Camera Calibration

To ensure accurate 3D trajectory estimation of the shuttlecock using Epipolar geometry, camera calibration was necessary to determine the intrinsic properties of the cameras. This process involved printing a checkerboard image generated online[3] on Letter Size paper and affixing it to the back of a movable tabletop stand. The cameras were mounted on tripods in fixed positions while the movable stand was translated to capture images of the checkerboard from various angles. These images were then input into the ”Camera Calibrator” App on MATLAB, which calculated the camera parameters, including intrinsic properties.

On Table 1 are the camera parameters of the two smartphones used in generating the dataset.

Ultimately, the calibrations of this specific camera set was not used since we the video footage shot were not suitable for evaluation for reasons described in section 4.

3.3. Shuttlecock Detection

Two approaches were tested to detect the Shuttlecock. The first approach uses background subtraction and the second approach uses the YoloV5 model. Sections 3.3.1 and 3.3.2 are the details of both approaches. We ultimately decided to use the deep learning approach using the YoloV5 model.

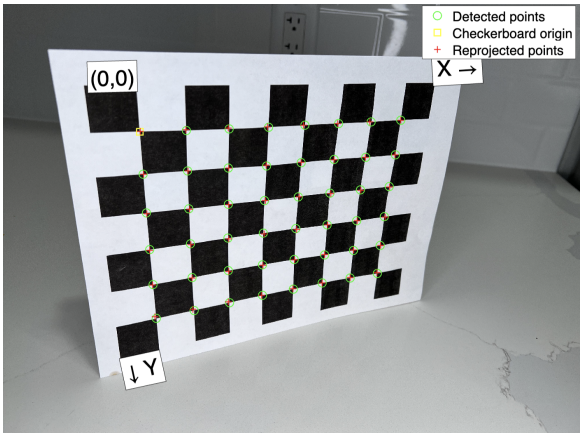

Figure 4. Camera Calibration using MATLAB “Camera Calibrator App”

Parameters	Camera 1	Camera 2
Fx	3149.102	2951.122
Fy	3148.456	2989.857
Principal Point x	2012.223	2007.837
Principal Point y	1457.441	1388.993
Radial Distortion x	0.214	0.267
Radial Distortion y	-0.703	-0.887
Tangential Distortion x	0	0
Tangential Distortion y	0	0
Skew	0	0

Table 1. Intrinsic Parameters for Both Cameras used in generating Dataset

3.3.1 Background Subtraction

The first approach was using background subtraction. We input a video file and apply background subtraction as we cycle through the image frames of the file. The goal is to detect consistently the moving blob in the image which we will assume is the shuttlecock; we do this by using a Gaussian Mixture Model (GMM). The GMM algorithm models each pixel in a video sequence as a mixture of Gaussian distributions, which represents the probability distribution of pixel intensities over time. The background model is built over a certain number of frames, and then the foreground is identified by comparing the current frame with the background model. However, after applying the model and cycling through a number of frames of the badminton game, we notice that there is a lot of noise (random movement from the crowd and other particles in the frame) that makes it difficult to accurately determine which blob belongs to the shuttlecock. Figure 4 shows an example frame with noise.

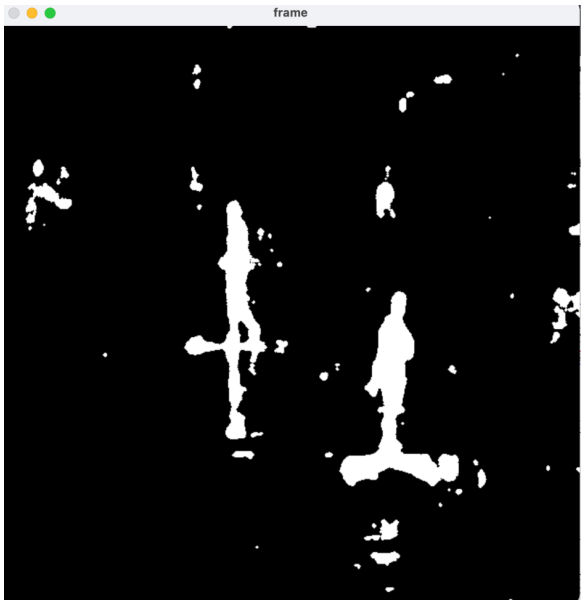

Figure 5. Shuttlecock detection using Gaussian Mixture Model

3.3.2 Shuttlecock Detection using YoloV5

The YOLOv5 algorithm was selected for its efficiency and accuracy in object detection tasks. The architecture consists of a convolutional neural network that is trained to predict the bounding boxes and class probabilities of objects in an image. The YOLOv5 algorithm was implemented using PyTorch and trained on a high-performance computing cluster.[2]

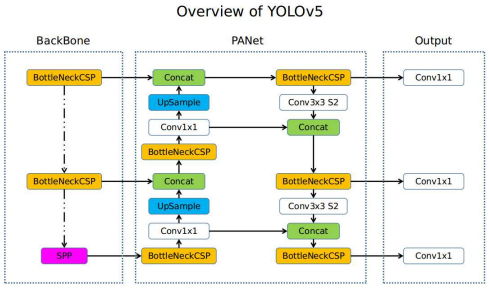

Figure 6. YOLO v5 Model Architecture

3.4. Extraction of the boundaries of the court

To achieve this, we utilized the opencv library and Python programming language. In this paper, we discuss the steps involved in this method, including image preprocessing, line detection, and boundary extraction.

Image Preprocessing: The first step in the process is to calculate the luminance of the input image. This is achieved by computing a weighted sum of the blue, green, and red channels. The luminance is used to detect the white lines in the image. We use a threshold value and a difference threshold to determine the presence of the white lines.

Line Detection: The standard Hough transform algorithm in the opencv library is used to detect lines in the image. The detected lines are then filtered to remove those that do not belong to the court. This is achieved by analyzing the slope and position of the lines relative to the image.

Boundary Extraction: The intersection points of the filtered lines are then utilized to determine the possible court boundary coordinates. We then extract the four coordinates at the edges of the court. With these four coordinates, we are then able to draw lines to indicate the boundary lines of the court. The extracted boundaries are then used for further game analysis to determine if the shuttlecock is within or outside the court boundary.

Figure 7. Image Preprocessing

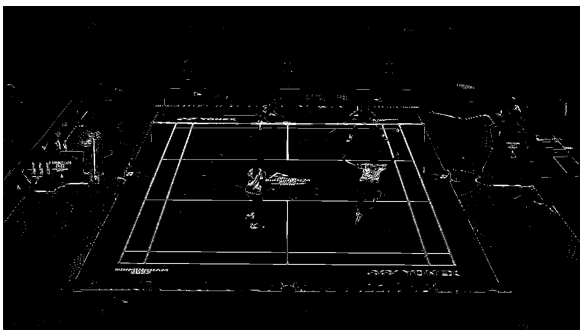

Figure 8. Line Detection

The green lines in Figure 9[1] are the detected boundaries of the court. From the image, we can see that we are currently detecting approximately 4 correct corners with some error.

[1] Seen on next page

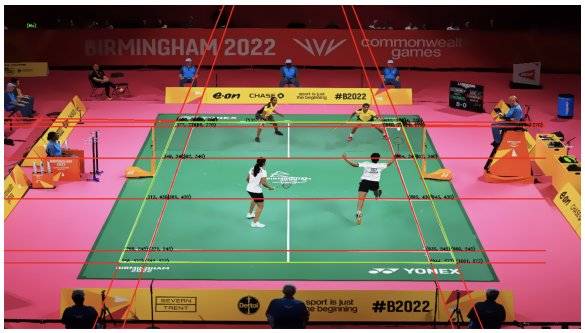

Figure 9. Boundary Extraction

3.5. Trajectory detection using Kalman filters

For trajectory estimation, we designed our own Kalman Filter which has the ability to perform estimations in both 2D and 3D coordinates with the help of a toggle. Taking a sequence of frames and extracting the position vector of the shuttlecock over time, we input that to the Kalman Filter to get predicted shuttlecock positions over time through the image frames. We compare the results of the Kalman filter with the actual measured results. For greater performance comparisons, we compare our Kalman Filter performance on the measured data with the performance of pythons filterpy Kalman Filter library and we outperform the library for shuttlecock position estimation on a badminton court. Table 3 shows the comparison of our Kalman Filter to Python Library for this task.

3.6. Determine if the shuttlecock is within or outside

Although the intention at the start of the project was to estimate the position of the shuttlecock using 3D estimation, due to limitations caused by our dataset discusses in section 5.2 we had to detect if the shuttlecock is in or out using 2D estimation by utilizing the ray casting algorithm.

The ray casting algorithm is a technique used to determine if a point lies inside a polygon or not. The algorithm works by casting a ray from the point in question in a direction and then counting the number of times the ray intersects the edges of the polygon. If the number of intersections is odd, the point is inside the polygon. If the number of intersections is even, the point is outside the polygon.

In the context of our project, the court boundaries were used to define a polygon with four vertices as a list of tuples, representing the four corners of the court. When a shuttlecock lands, its coordinates are recorded, and then the ray-casting algorithm can be used to determine whether the shuttlecock is inside or outside the court. We return True if the shuttlecock is inside the court, and False if it is outside.

Figure 10. Court Boundary Detection

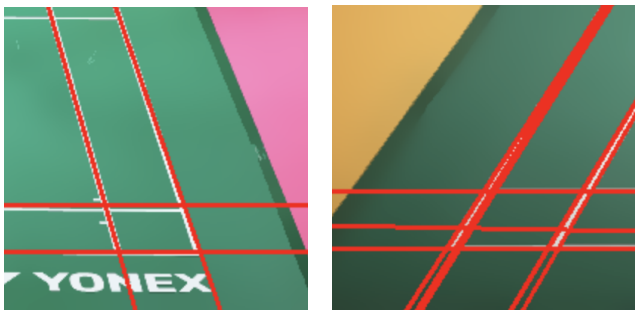

Figure 11. Normal (Left) vs Inaccurate (Right) Boundary Detection

4. Evaluation

4.1. Court Boundary Detection Accuracy

To assess the precision of the court corner-finding stage, we conducted tests not only on the court used in our experiment but also on additional images of badminton courts obtained from the internet. These images were chosen as appropriate tests for our algorithm since they were captured in well-lit conditions with the entire court in view. The court utilized in our experiments, along with the identified boundary locations annotated with boundary markers as red lines, as presented in Figure 10. In every badminton court image tested, the court's corners were detected with precision, albeit with minor discrepancies in their positions shown in Figure 11 by the presence of double lines which makes it more difficult to automatically choose the right intersection. To assess accuracy, we calculated the distance error in image-space based on the known positions of the boundary to obtain accurate measurements of the precision of corner detection. Table 3 presents the mean absolute error for the detected boundary of identified corners in each input image.

4.2. Shuttlecock Initial Detection Model Accuracy

After employing the YoloV5 model, illustrated in Figure 6, for training purposes, we achieved a prediction accuracy

of 68.6%. However, the recall value and mAP were 42.2% and 44.6% respectively. Despite attempts to enhance accuracy by expanding the data set, there was no substantial improvement in accuracy. Figure 12 portrays the model's prediction outcomes, with the yellow bounding box indicating the predicted position of the shuttlecock and its corresponding confidence value OF 63%.

Figure 12. Shuttlecock detection using Model

mAP	precision	recall
44.6%	68.6%	42.2%

Table 2. Result of YoloV5 model training

4.3. Shuttlecock Tracking Accuracy

Table 3 and Figure 13 shows Kalman Filter performance for a badminton rally with over 12 position measurements of the shuttlecock.

Parameters	Our KF	Python Lib KF
Accuracy within 10 pixels	91.67%	83.33%
Average Error (pixels)	4.5	4.9

Table 3. Kalman Filter accuracy

For the final test with the Kalman filter superimposed on the test image. We ran the test over a rally of 34 image frames. Figures 14 and 15 show the Kalman filter prediction vs the measured output on the original image.

4.4. Shuttlecock+Line Call Accuracy

Due to the fact that we are working in 2D and single camera views, it is difficult to be truly accurate with the out-of-bounds detection. Our accuracy for determining if the shuttlecock is in or out of bounds is low 60%. Figure 16 below shows the line call.

The two black ellipses in Figure 16 indicate the boundary call (top left) and the shuttle cock position at the time of the boundary call (center-left). In this image, the call is 'Yes' which is correct.

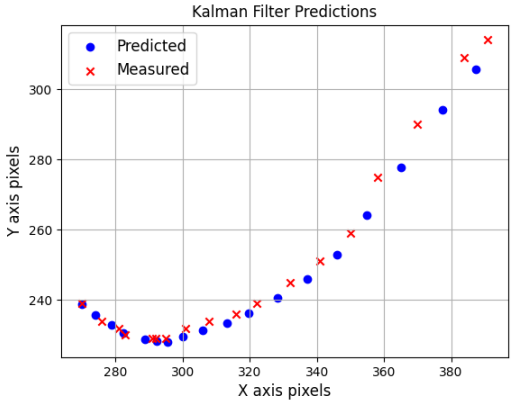

Figure 13. Kalman Filter Output

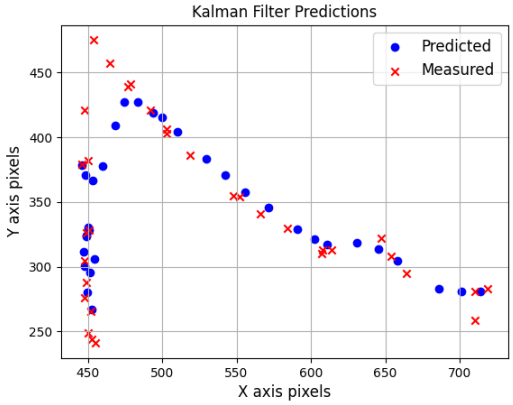

Figure 14. Kalman Filter Output final sequence

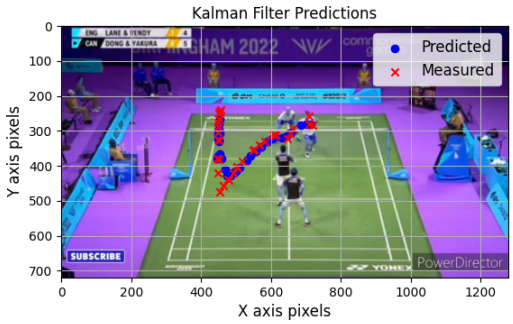

Figure 15. Shuttlecock predicted trajectory over time

On Figure 9[2] are graphs of our model training.

[2]Seen on next page

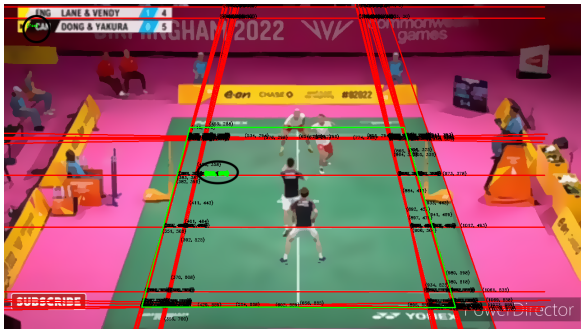

Figure 16. Shuttlecock boundary call

Figure 17. Multiple shuttlecocks and courts

5. Discussion

5.1. Limited Data Capture

In attempting to reproduce this paper, we used a lot of the original dataset, and further annotated additional images. Furthermore, we attempted to create our own dataset but ran into difficulties as discussed further in the next two sections.

5.2. 3D Position Estimation Difficulties

In attempting to generate our own dataset, we ran into a couple of complications and things we find important to be aware of as others attempt to do the same. Firstly, there exist some simplifications of note when thinking about the game of badminton when played and recorded officially. Firstly, there is a single shuttlecock in play, and no extras on the floor Figure 17 shows the multiple shuttlecocks in an image frame problem. This allows for the detection of the single object in play in order to better determine whether it is in or out of bounds. Besides this, there exists a single court in view and the boundaries can therefore better be determined. What might be of interest as well, could be the increased usage of manual understandings of space, particularly where court boundaries exist, so that the shuttlecock at every position is the only thing that needs identifying. The usage of multiple angles is additionally useful in identifying such. An additional difficulty is in the lower resolution images, the shuttlecock itself is incredibly difficult to find, leading to even greater difficulties when trying to generate the dataset with a shuttlecock that is blurry due to its speed, alongside an easily identifiable extra one on the ground.

5.3. Trajectory Estimation Difficulties

We encountered challenges in accurately estimating the trajectory of the object as we lacked a second camera angle for 3D estimation. Consequently, we had to rely on 2D estimations for tracking the object's movement in 3D space, which resulted in a degree of inaccuracy in the estimations. Hence why we have low accuracy on the shuttlecock in or out of bounds decision call.

5.4. Camera Calibration Difficulties

Regrettably, our recorded dataset did not include any locations where a single badminton court could be captured in view, and no suitable camera heights were available to obtain multiple views of the same scene. Consequently, we resorted to using a dataset from the televised Badminton Commonwealth Games, which only provided a single view of each scene. This limitation forced us to rely entirely on the image coordinate system, precluding the ability to convert to the originally intended world coordinate system.

5.5. Possible Improvement

An emphasis would exist on a greater set of data with annotation. Given the shorter duration, the dataset contained about 5000 images, but given a longer duration and ability to annotate with higher resolution images, the shuttlecock could be more easily identified by the human, before utilizing the yolov5.

Further improvements can be made by considering the estimation of the shuttlecock position in 3D rather than 2D. This can be achieved by utilizing videos of the same game captured through calibrated cameras from opposite sides of the court. However, capturing an unobstructed view of a single court without neighboring courts proved to be a challenging task for us, despite our attempts to do so as discussed in the section 3.1.

6. Conclusion

In conclusion, the objective of this study was to develop a method for detecting the trajectory of a shuttlecock and determining whether it landed within the bounds of a badminton court. Through the process outlined in this paper, we were able to successfully identify the court boundary and shuttlecock trajectories using stereo footage. Furthermore, we utilized the Yolov5 model to detect the initial position of the shuttlecock. Although our system is not fully functional, our results thus far demonstrate that shuttlecock tracking is feasible for amateur badminton players using position estimation. With additional refinement and more

time, this approach holds promise for enhancing the training and performance of badminton players.

7. Appendix

All team members were involved in all stages of the project, but each team member took the lead on the following tasks:

- Peter: Camera setup and calibration, video capture, court boundary detection and determining is shuttlecock was within or outside the court
- Bryan: Court boundary detection, Kalman Filter, Accuracy tests
- Yaron: YoloV5 model training and test, shuttlecock detection using YoloV5 model.

Additionally, you can find the codebase used in implementing this solution on Github [1]

References

[1] Codebase used to project technical approach. https://github.com/mbanefo2/CS231FinalProject. 8

[2] Yolov5 architecture. https://github.com/ultralytics/yolov5. 4

[3] M. Cartron. Shuttlecock dataset and pre-trained model. https://universe.roboflow.com/mathieu-cartron/shuttlecock-cqzy3/browse?queryText=split 2

[4] Swalaganata et al. Galandaru. Super-resolution imaging applied to moving object tracking, 2017. 1

[5] Vrajesh et al. Shah. Shuttlecock detection and fall point prediction using neural networks, 2020. 2020 International Conference for Emerging Technology (INCET), Belgaum, India,. 2

[6] Hidehiko et al. Shishido. A trajectory estimation method for badminton shuttlecock utilizing motion blur., 2013. Image and Video Technology. Springer Berlin Heidelberg. 1, 2

[7] Chen et al. Wei. Using ftoc to track shuttlecock for the badminton robot. 1