Baseball Pitch Identification Using Hough Circle Transform and Decision tree

Harold Fan
yucaif@andrew.cmu.edu

Alan Grizzaffi
alan.grizzaffi@west.cmu.edu

Grayson Wu
jiahaow@andrew.cmu.edu

Caleb Parks
cdparks@andrew.cmu.edu

Abstract

In sports, the proper use of data is key in gaining an edge over the competition for teams and individuals. Although this data is generally readily available to the public, the data collection methods tend to be kept hidden. Baseball must rely on machine learning methods to improve upon fan viewership through ball trajectory paths and pitch location identification. For this reason, we decided to analyze the MLB-YouTube dataset to see how well our pitch detection works as compared to the annotated broadcast notes.

In order to classify the pitches properly, we employ the use of Hough Circle Detection in order to create the trajectory path of the ball and then use XGBoost for the actual classification method. To create the trajectory, entire videos were segmented into clips containing individual pitches, then broken down into usable frames that included only the pitch itself. These frames are then filtered using GaussianBlur and transformed into binary images by subtracting adjacent frames. Hough Circle Detection is used to locate the ball and thus the trajectory path. Then, the trajectory is fit using a polyfit function and is input into XGBoost along with the JSON file created containing other pertinent pitch information as well.

We created a gradient boosted CART decision tree with a maximum depth of 6 and an objective function of softmax. The softmax function was chosen due to the multi-class nature of our problem. Next, we trained the model using 370 pitches and tested on 52 pitches. Lastly, we shuffled the dataset and repeated training 50 times which ended with us selecting the model with the highest averaged F1 score on the 52 pitches. This model achieved an and averaged F1 score of .96 on training data and .82 on a testing set of 250 pitches.

1 Introduction

Pitch detection is probably one of the most important statistics in the game of baseball, as it allows for pitchers and catchers to construct game plans against opposing hitters who may struggle against certain pitch types and locations. This analytic leads team managers to construct certain batting lineups, fielder's positions, and much more based on what the pitcher tends to be throwing. For example, pitchers who tend to throw slower with better command of their off-speed pitches (curveballs, changeups, sliders, etc.) can typically be known as "ground ball" pitchers, meaning the infield would need to be positioned properly in order to reduce the chances of a hit from the opposing team. These metrics come from Statcast analysis taken on every batter and the player's respective spray charts. It is this constant game of "data analytics chess" that has began leading to a lot of the new developments in the game of baseball, such as the defensive shift or improved optimum swing paths (increasing the run producing potential of every batter). Today, the MLB uses PITCHf/x

32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.

and Statcast systems that employ multiple high-speed cameras and radar in order to classify pitch trajectory and selection. As stated above, the collection methods are typically not disclosed, so we will be analyzing the MLB-YouTube dataset and determining how our pitch classification methods hold up against that of the MLB. We will use the densely annotated frames from the database specified in order to detect pitch types, from the single broadcast view that all fans see on the television. This will be done by employing a combination of Hough Circle Transform and XGBoost to track the baseball's trajectory and ultimately determine the pitch type through a proper learning algorithm. No use of neural networks will be incorporated in our pitch detection methods. The goal of this project is to be able to properly classify the pitch type with comparable accuracy to that of the MLB's undisclosed algorithms.

2 Related Work

2.1 Hough Circle Transform[1]

If a 2D point (x,y) is fixed, then the parameters can be found according to equation. The parameter space would be three dimensional, (a, b, r), and all the parameters that satisfy (x, y) would lie on the surface of an inverted right-angled cone whose apex is at (x, y, 0). In the 3D space, the circle parameters can be identified by the intersection of many conic surfaces that are defined by points on the 2D circle. This process can be divided into two stages. The first stage is fixing radius then find the optimal center of circles in a 2D parameter space. The second stage is to find the optimal radius in a one dimensional parameter space. We can use Hough Circle Transform to detect the ball in each frame, then track the trajectory for further use.

2.2 Polynomial Curve Fitting

Curve fitting is the process of constructing a curve, or mathematical function, that has the best fit to a series of data points, possibly subject to constraints. Curve fitting can involve either interpolation, where an exact fit to the data is required, or smoothing, in which a "smooth" function is constructed that approximately fits the data. Fitted curves can be used as an aid for data visualization, to infer values of a function where no data is available, and to summarize the relationships among two or more variables. Extrapolation refers to the use of a fitted curve beyond the range of the observed data, and is subject to a degree of uncertainty since it may reflect the method used to construct the curve as much as it reflects the observed data. [2]. We can use Polynomial Curve Fitting to get a smooth trajectory of a scatter trajectory and, at the same time, the feature of this trajectory.

2.3 Decision Tree and XGBoost

A decision tree is a decision support tool that uses a tree-like model of decisions and their possible consequences including chance event outcomes, resource costs, and utility. It is one way to display an algorithm that only contains conditional control statements. Decision trees are commonly used in operations research, specifically in decision analysis, to help identify a strategy most likely to reach a goal, but are also a popular tool in machine learning[3].

The Classification And Regression Tree (CART) analysis is an umbrella term used to refer to both of the above procedures, first introduced by Breiman et al. in 1984. Trees used for regression and trees used for classification have some similarities - but also some differences, such as the procedure used to determine where to split.[4].

XGBoost is made of decision tree ensembles, which consist of a set of decision trees (typically CARTs) using a regularized term to optimize fitting results. In our project, we used XGBoost to judge the pitch based on features extracted from trajectory.

[1] https://en.wikipedia.org/wiki/Circle_Hough_Transform
[2] https://en.wikipedia.org/wiki/Curve_fitting
[3] https://en.wikipedia.org/wiki/Decision_tree
[4] https://en.wikipedia.org/wiki/Decision_tree_learning

3 Method

3.1 Data Processing

After we get our dataset, which are entire videos of several baseball game recordings, we use ffmpeg[5]. to segment the game into short clips showing only one pitch at a time. Handling the entire video is not convenient, as we need the video to be broken down into frames in order to obtain the trajectory and ultimately classify the pitches. Thus, we used the cv2 module to capture the video at a fixed sample rate to acquire a series frames of each pitch. Next, we used the PIL module to obtain binary frames showing the movement of objects between adjacent frames.

3.2 Circle Detection

The feature we need to collect is the baseball location in every frame. Since the ball can be seen as a circle, we decided to give up on using SVD to detect the ball as it seemed that circle detection would be more fitting. Thus, we chose to use Hough Circle Transform to detect to the ball in each frame, as it is faster and easier to use in regards to parameter tuning. Before attempting the circle detection, we specified a search area in which the ball is expected to be and set pixel values of unreasonable areas to zero. The reasoning behind filtering the frames was to remove extra "noise" that could reduce our accuracy during detection while potentially increasing computation time. Cropping of the photos included setting pixel values of all unnecessary information (where ball logically could not be) to zero and leaving the essentials; the pitcher, the path of which the ball travels down (pitch channel), and the catcher/umpire. GaussianBlur was applied to all frames prior to binarization in order to "smooth" the images and remove extra noise that could affect the detection method.

3.3 Trajectory Classification

After acquiring multiple positions of the ball in sequential frames, we were able to stitch them together to determine the pitch trajectory. We used a polynomial curve fitting to turn a scatter trajectory formed by a series frames into a smooth curve and its functional expression, which can be treated as its feature. Other features utilized were the start and end points of this trajectory, the total number of frames (used to indicate the speed of this pitch), and the handedness of the pitcher. Due to the release point, the same pitch type will have a different trajectory from left-handed and right-handed pitchers when being observed from the same camera angle. Finally, by using XGBoost, the decision can be based on all features, layer by layer, and then added to a softmax function before final classification.

4 Dataset

In this project we are using MLB-YouTube, designed for fine-grained activity detection. It's a large-scale dataset consisting of 20 baseball games from 2017 MLB post-season, which is available on YouTube with over 42 hours of video footage. The dataset contains two settings: segmented video classification as well as activity detection in continuous videos. The motion and appearance difference between the various activities is quite small, so it may be an advantage for us because we avoid complex context in our task. Although it meant to be a dataset used for video activity recognition and classification, because this dataset has labels of pitch type of each video segment, we can take advantage of it, where we can use it to do baseball tracking and pitch type classification. MLB-YouTube segmented video dataset consists of 4,290 video clips. Each clip is annotated with the various baseball activities that occur, such as swing, hit, ball, strike, foul, etc. A video clip can contain multiple activities, so it requires additional work to do if we want to make sure there is no more than one pitch occur in a clip.

The individual frames will largely be identical to each other in terms of pixel values. Most of the frame will be values very close to zero or zero (black) with the ball being white. Depending on the team, the players may also be wearing colors close to white and this may effect edge detection. From these frames, we plan on extracting how many frames the ball moved in both the X and Y direction, how fast it moved based on frames, and other features to classify pitches.

[5]https://www.ffmpeg.org/

5 Results

5.1 Video Clip

The whole game video could be segmented to a set of frames, as shown in Figure 1 below.

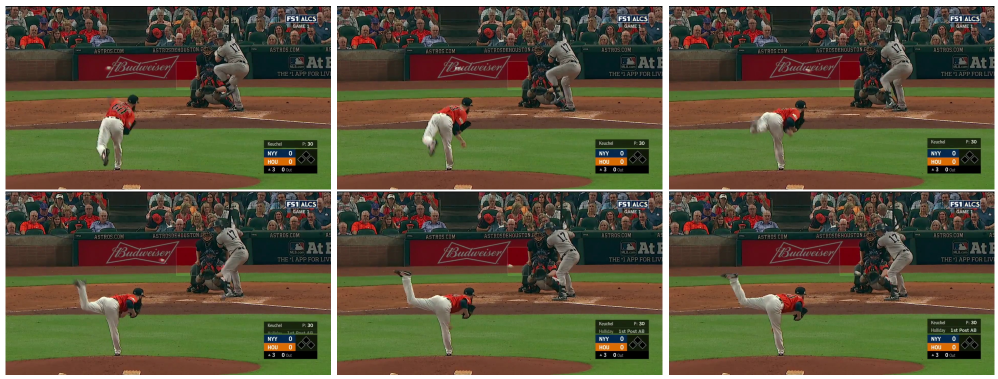

Figure 1: One pitch in 30 fps

5.2 Feature Extraction

Then, we subtract adjacent frames pixel values and filter below a certain threshold to create a set of binary frames to aid the Hough Circle Transform during detection, as shown in Figure 2 below.

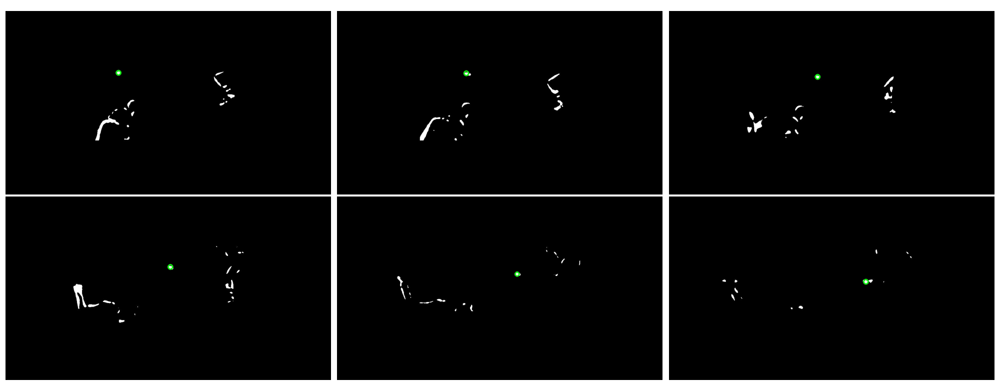

Figure 2: Blob and ball detection

Next, we merged all detected balls into one frame to get a scatter trajectory. With this trajectory a polynomial curve fitting was used to simulate a smooth trajectory to obtain the parameters of the fit function, as shown in Figure 3 below.

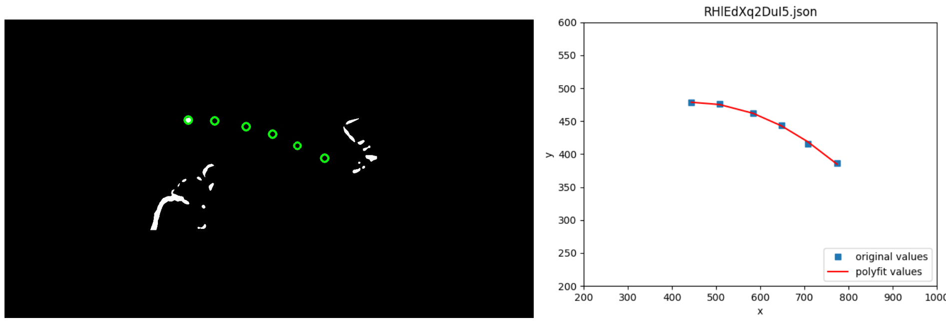

Figure 3: Scatter and fit function of trajectory

5.3 Classification

Around 90% data was used as training data where the w_0, w_1 and w_2 features of the fit function, start and end point, left or right hand, number of frames, and the label, were used to update the decision tree. Then, extracted features from the test data and decision tree gave us the results as per Figure 4 below.

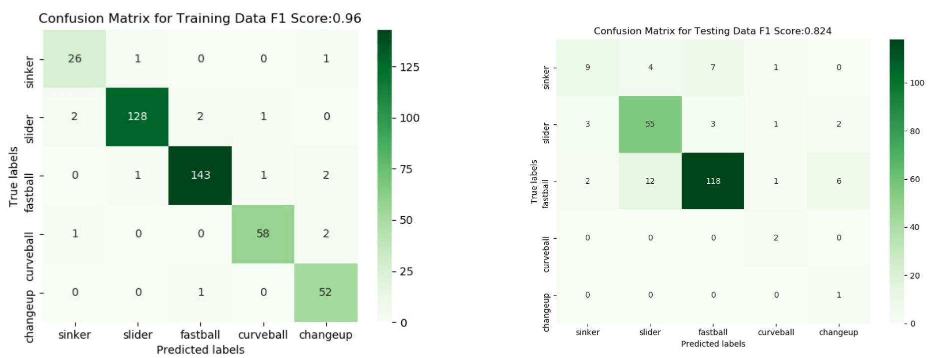

Figure 4: Confusion Matrix for training data and test data

6 Discussion and Analysis

Evaluation of the results before the model took place in two parts. First, the detection results had to be evaluated. Our method was not able to track the ball in every frame, rather it only needed to capture the arc of the ball to be effective. So, when evaluating the results we had to visually inspect frames of pitches to see if the arc was adequately captured. For classification results after detection, we used the F1-score and accuracy (confusion matrix) as our metrics for each pitch type, as shown in Figure 4.

The dataset has timestamps and labels for every pitch of each game. Those labels were used to test our accuracy on a test set of data. As we can see from Figure 5, the most important feature is start point while the least important was w_1 which indicate the position of the curve. What's more, the hand information is not in this figure because the hand information have already exist in the start point thus it was not in any of the decision trees.

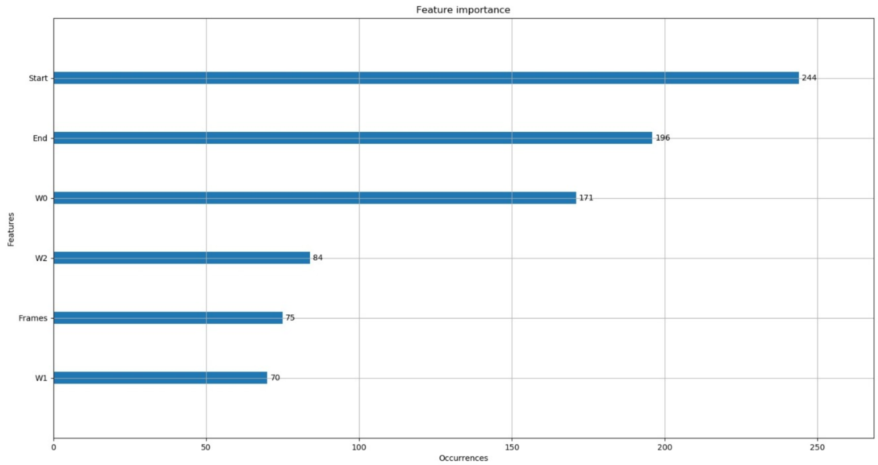

Figure 5: Feature analysis

For limitations, we did not have a extensive dataset. This is mainly because frames of pitches that were able to be used by the model had to be extracted manually which is very time intensive. By using more data, we could have made our model more robust. Another limitation we realized, was the lack of cross-validation. In the future, we would want to divide the data based on the pitcher, in order to avoid using pitches from the same pitcher in the training and testing datasets.

References

[1] AJ Piergiovanni and Michael S. Ryoo. "Fine-grained Activity Recognition in Baseball Videos" arXiv:1804.03247 (2018).

[2] Perera, Asanka G; Law, Yee Wei; Chahl, Javaan. "UAV-GESTURE: A Dataset for UAV Control and Gesture Recognition" arXiv:1901.02602 (2018).

[3] Kay, Will; Carreira, Joao; Simonyan, Karen; Zhang, Brian; Hillier, Chloe; Vijayanarasimhan, Sudheendra; Viola, Fabio; Green, Tim; Back, Trevor; Natsev, Paul; Suleyman, Mustafa; Zisserman, Andrew. "The Kinetics Human Action Video Dataset" arXiv:1705.06950 (2017).

[4] Kingma, Diederik P.; Ba, Jimmy. "Adam: A Method for Stochastic Optimization" arXiv:1412.6980 (2015).

[5] Zeiler, Matthew D.. "ADADELTA: An Adaptive Learning Rate Method" arXiv:1212.5701 (2012).

[6] Kang, Soo Min; Wildes, Richard P. "Review of Action Recognition and Detection Methods" arXiv:1610.06906 (2016).

[7] AJ Piergiovanni, Michael S. Ryoo. "Unseen Action Recognition with Multimodal Learning" arXiv:1806.08251v2 (2018).

[8] J. Carreira and A. Zisserman. "Quo vadis, action recognition? A new model and the kinetics dataset" arXiv:1705.07750v3 (2017).

[9] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. "Rethinking the inception architecture for computer vision" In Proceedings of the IEEE Conference on Computer Visionand Pattern Recognition (CVPR), 2016

[10] A. Piergiovanni, C. Fan, and M. S. Ryoo. "Learning latent sub-events in activity videos using temporal attention filters" arXiv:1605.08140v3 (2017)

[11] A. Piergiovanni and M. S. Ryoo. "Learning latent super-events to detect multiple activities in videos" arXiv:1712.01938v2 (2018)

[12] A. Piergiovanni and M. S. Ryoo. "Temporal Gaussian Mixture Layer for Videos" arXiv preprint arXiv:1803.06316v6 (2018)

[13] H. Andrews and C. Patterson. "Singular Value Decomposition (SVD) Image Coding" DOI: 10.1109/TCOM.1976.1093309 (1967)

[14] Yoav Freund and Robert E. Schapire. "Experiments with a new boosting algorithm" ICML'96 Proceedings of the Thirteenth International Conference on International Conference on Machine Learning, Pages 148-156, Bari, Italy — July 03 - 06, 1996

[15] https://www.youtube.com/user/MLB

[16] http://mlsp.cs.cmu.edu/courses/fall2019/lectures/lecture7_boosting_only.pdf

[17] Wikipedia contributors. "Circle Hough Transform." Wikipedia, The Free Encyclopedia. Wikipedia, The Free Encyclopedia, 17 Aug. 2018. Web. 7 Nov. 2019.

[18] Wikipedia contributors. "F1 score." Wikipedia, The Free Encyclopedia. Wikipedia, The Free Encyclopedia, 5 Nov. 2019. Web. 7 Nov. 2019.