Basketball Shot Prediction Using Neural Networks

Justin Wolkowicz
Boston College
wolkowij@bc.edu

Abstract

Computer vision applied to the realm of sports has created various methods of automation. In particular, the implementation of shot prediction and ball tracking automates many components within sports, including sports analytics and broadcasting. This project sought to construct a such a model that focused entirely on basketballs and their shot trajectories. The approach utilized the YOLOv5 neural network for basketball detection and a polynomial of best fit over the positional data of the detected object. The detection component of the model achieved high average precision (97%) on the test set, and coupled with polynomial regression succeeded in forming a naive shot predictor.

1. Introduction

The ability to track and predict components of a game has been an objective throughout the history of sports. The field of computer vision offers many methods that seek to automate this goal. Specifically, basketball is a sport that can benefit extensively from this type of automation. In the fast-paced game, it can be difficult to manually track and predict the movement and trajectory of basketballs. As such, implementing a model that automates the trajectory prediction of basketball shots is beneficial to those in the sports industry. Further, a performant model for trajectory prediction is of particular interest to those in sports analytics. Therefore, the objective of this project is to implement a computer vision model that can accurately predict shot trajectories in basketball videos.

In order to implement such a model, I will utilize the YOLOv5 neural network applied to basketball detection [1]. Its architecture allows for single-pass object detection in each frame, which will be essential in efficiently performing inference on the data. The trajectory predictions will be polynomials of best fit regarding the positions of the basketballs between multiple frames. With these components combined, the model should be able to accurately plot shot arc predictions within a video.

Common with many computer vision projects, trajectory prediction will be a challenging task. Restriction of camera angle, visual interference, and player obstruction can all cause problems that could hinder the performance of the detection algorithm. Therefore, this project will attempt to identify and lessen the effects of these potential problems.

2. Related Work

Throughout the field, there have been many efforts in implementing trajectory prediction models applied to sports videos. The objectives range from predicting player movements to predicting the motion of game pieces such as tennis balls or basketballs. For instance, the approach of Maksai et al. [5] was to model the motion of a volleyball through representing the path as a graph associated to various states of the ball. In doing so, the authors were able to encode functions that could predict the state and position of the ball at future times. This approach is elaborate, and likely is computationally expensive. Furthermore, it requires additional information including player detection and action classification. If this approach were to be replicated, it would require much more involvement of the detection model.

The work of Acuna [2] was focused on player detection using a YOLOv2 detector and motion tracking using a Kalman filter framework. This enabled the author to estimate velocities based on the data from the current and previous frame, which is very efficient. The author also proposed a method for motion tracking when object detection was interrupted through utilizing a linear velocity model based on data from previous detections. This approach, although applied to player movements, would translate well to mapping the motion of basketballs.

Another approach [3] uses detection positions to estimate both two-dimensional and three-dimensional trajectories of basketball shots. The method applies physics equations on the set of local position points on the ball to calculate directional velocities and estimate depth. Most importantly, it uses a best-fitting function approach to estimating the motion of the ball. An implementation of this approach would likely require inference on multiple frames, or perhaps multiple cameras, to estimate the three-dimensional

positions of the balls, which could be inefficient. However, the "function of best fit" approach to estimating trajectory would be a very useful method applied to this project.

3. Method

In implementing the shot prediction model, I elected to use a YOLOv5 object detector where trajectories were based on the positions of the detected object. All stages of the pipeline were done in Pytorch, as it was the configuration of the dataset and the model repository. The dataset I used was the *Basketball* dataset [4] which contained around 2600 images of basketballs in various settings, along with 2600 bounding box coordinates associated to each basketball. All data was preprocessed into 416×416 resolution and split into a $77\%/5\%/18\%$ train/validation/test split.

3.1. YOLOv5

Upon obtaining the split dataset, the data was applied to the YOLOv5 model for training.

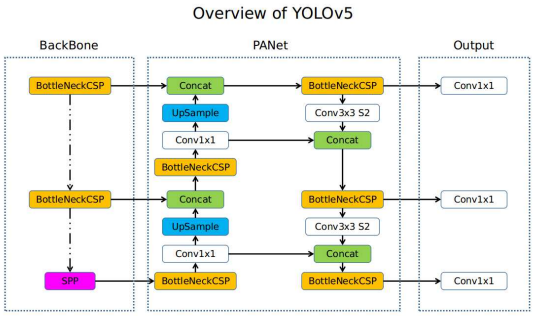

Figure 1. YOLOv5 Architecture [1]

The model architecture displayed in 1 contains two distinct components: the BackBone and PANet. The layers within the BackBone of the YOLOv5 architecture are intended to extract informative features from the input. As for PANet, the layers generate feature pyramids that allow for classification invariant of scale and size of the input. In training this network to detect basketballs specifically, I elected to train every layer of the network. The training process consisted of 100 epochs with input batches of 16.

Within the training process, loss was calculated through a combination of three metrics: box loss, classification loss, and objectness loss. Box loss is simply the mean-squared error between the predicted and true bounding box center. Classification loss is the cross entropy loss between the label and the predicted class. Objectness loss is the cross entropy loss between the predicted bounding box's confidence value and the ground truth (1).

Upon the completion of training, model performance was assessed using precision-recall scores on the test data.

3.2. Polynomial Regression

The trajectory prediction step was done through an augmentation of the YOLOv5 source code, namely their inference file *detect.py*. This process stored the positions of up to 10 consecutive detected basketball positions and on each frame calculated the two-degree polynomial of best fit through these points. The polynomial was then displayed to the individual frames using OpenCV. The following result led to a continually updating polynomial of best fit throughout the video.

4. Experiments

As detailed in the previous section, the experiments were done on the *Basketball* dataset [4] with a training process consisting of 100 epochs of batch size 16. Since the data was already preprocessed into 416×416 resolution images, training was relatively simple in approach. As the training process was computationally expensive, I utilized the free GPUs given on Google Colab. The optimal detection model determined by the training process was evaluated on the test set using precision-recall scores. As a baseline, I used the player detection performance detailed by Acuna's paper [2]. In those results, the trained YOLOv2 models obtained an upper bound of about 90%. Therefore, the relative performance of my model will be evaluated on a baseline of around 90% precision.

Class	Precision	Recall	mAP50	mAP50-95
Basketball	0.918	0.947	0.971	0.716

Table 1. YOLOv5 Model Results

As shown, the YOLOv5 model trained under the parameters listed above achieved very high precision and recall scores, indicating that the detection on the test set made very little mistakes in classifying basketballs. Furthermore, the higher recall value indicates that the model preferred to misclassify non-basketball objects as a primary source of error. Relative to the chosen baseline, this model achieved very decent performance in classifying basketballs.

Figure 2. Detection examples on the test set. All basketballs are detected in this batch, even in the noiser and darker images.

In contrast to the object detection model, the trajectory prediction component does not have a ground truth. Therefore, the evaluation of the predictions would not need to be evaluated on the same test set, and did not have an associated scoring metric. Due to this, I elected to evaluate the trajectories based on a simple eye test. The model was tested on a small batch of random out-of-sample basketball videos found online.

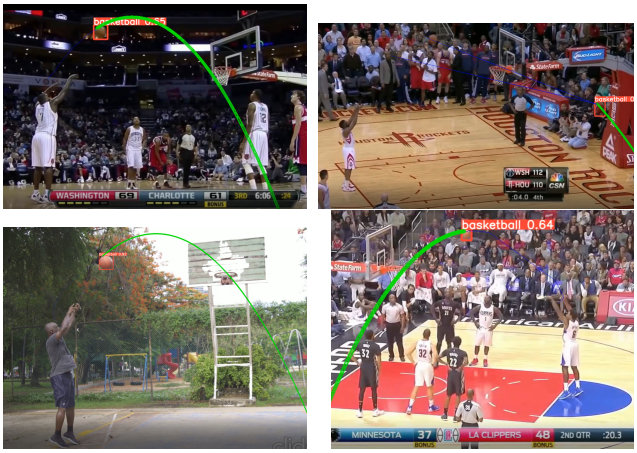

Figure 3. Collection of trajectory results from randomly sampled videos.

As shown by the figure 3, the results can be relatively accurate based on the position in time of the basketball. For instance, the trajectory position towards the beginning of the video often do not have enough positional data to plot accurate trajectories, but once the ball moves for a small amount of time, the prediction quality improves. The issues with this method occur when the object detection fails, as the plot either abruptly shifts upon receiving a misclassified object's position or cuts out entirely. Therefore, the performance of the trajectory prediction model entirely falls on the consistency of the YOLOv5 component, which seems to struggle on noisy out-of-sample data.

5. Conclusion

The results indicated that the YOLOv5 object detector combined with polynomial regression ultimately accomplished the original goal of this project. As shown in 3, the performance varied, likely due to the nature of the dataset and the lack of regularization in the process of fitting the polynomial. The dataset itself did not contain enough samples to be considered representative of any average image containing a basketball. Therefore, it would be beneficial to the out-of-sample performance of the model to retrain on a larger, more representative dataset of basketballs. The methodology of this project was presumably too simple to create a practical model, and more effort towards improving the consistency of the trajectory predictions through various methods (sample inliers, apply a Kalman filter) would achieve far better results. However, with detection comparable to other works [5], [2], this approach certainly succeeded in many parts. Although the final model may not be applicable to real-world problems, it remains as a proof-of-concept that could be extended to such.

6. Contribution

https://github.com/jwolk2/Basketball-Shot-Prediction

References

[1] https://docs.ultralytics.com/. 1, 2

[2] David Acuna. Towards real-time detection and tracking of basketball players using deep neural networks, 2017. 1, 2, 3

[3] Hua-Tsung Chen, Ming-Chun Tien, Y-Wen Chen, Wen-Jiin Tsai, and Suh-Yin Lee. Physics-based ball tracking and 3d trajectory reconstruction with applications to shooting location estimation in basketball video, 2009. 1

[4] EagleEye. Basketball image dataset. https://universe.roboflow.com/eagle-eye/basketball-1zhpe/dataset/1. 2

[5] Andrii Maksai, Xinchao Wang, and Pascal Fua. What players do with the ball: A physically constrained interaction modeling. 1, 3