Improving Quarterback mechanics through football tracking

Obsa Aba-waji

Abstract: In this report, we will be tracking a football in a video. The purpose of this is to improve the throwing mechanics of a quarterback. We will be looking at two different ways to track the football by looking at different ways to detect the football. The first will be using the color of the football, which is generally brown and uniform and the second way will be throw deep learning.

1. Problem Statement

We want to find a way of helping improve a quarterback's throwing mechanics. The quicker and tighter a quarterback's throwing motion, the more time they have to process the field as well as the less time defenders have to react to their throw.

2. Approach

I knew from class that object tracking had multiple parts, so I first split the problem into two sections. The first was the detection and the second part was the tracking.

For tracking, I had the idea to just simply find the center of the detected football, then store all the points in an array with a fixed size. The fixed size is because otherwise all the points will cover the entire frame. Then I draw a line that connects all the points together and show them on the frame.

For object detection, I decided to employ two approaches. the first approach was to use the color of the football. The reason is because footballs are generally uniform in color, so if I could detect the brown color, I could easily detect the football.

The second approach was to use deep learning. My plan for this was to use cv2's dnn module, which allows you to use darknet based models as long as you have the config and weights for the model.

3. Experiment Result

For my first detection, I was able to successfully find and track the football. I was also able to show the football being tracked as well as its path. However, because I was just using the a color range for tracking, I noticed that if i expanded the range too much, I would get false positives and I noticed that if I went more narrow with the range, it would not track the football as a whole, but track points on the football. This led me to changing my approach. I didn't believe I would get better results by continuing to tweak the color range.

My next approach was to use deep learning and neural networks to detect the football, then used the center of the detection rectangle as the points to be passed to my tracking function. I decided to use the pretrained YOLOv3 model. My reasoning was because YOLO based algorithms are supposed to be able to do real time detection, because they are only looking once and the data was pretrained on the COCO dataset, which has a category called sports ball. Being based on a real time detection model would mean a future version of my project could be used by teams in near real time when doing training or during the game to make slight adjustments.

The first problem I had with this approach was that the full version of yolov3 was processor intensive even with the data already pretrained. My computer was unable to handle it, which made it difficult to know if the error was in the model or if my computer was just unable to do the processing task. I switch to the lighter weight tiny-yolo. It was also pretrained on the COCO dataset. My computer was better able to handle this model, but I noticed that my output didn't show the football being highlighted. I removed my if statement, which only drew detections around sports balls to include all objects detected and was able to detect people.

This led my to hypothesize that because all the other sports balls in the image dataset were round that maybe it one of the weight in the model for sports balls was roundness. To verify my hypothesis, I found a high quality image of a football and passed it through to the model and the resulting detection was for a baseball glove.

4. Conclusions

While I was able to build a program that tracks a football in a video based on color, I was unable to do so currently with the deep learning approach. I initially believed I could find another pre labeled dataset to use and would only need to train on that dataset. I looked at the Open Image dataset and Imagenet. They both had football as a labeled dataset, but those were referring to soccer, so my next step will be to download photos of footballs then label all of the photos and create a trained model on that data.