Initial Concept

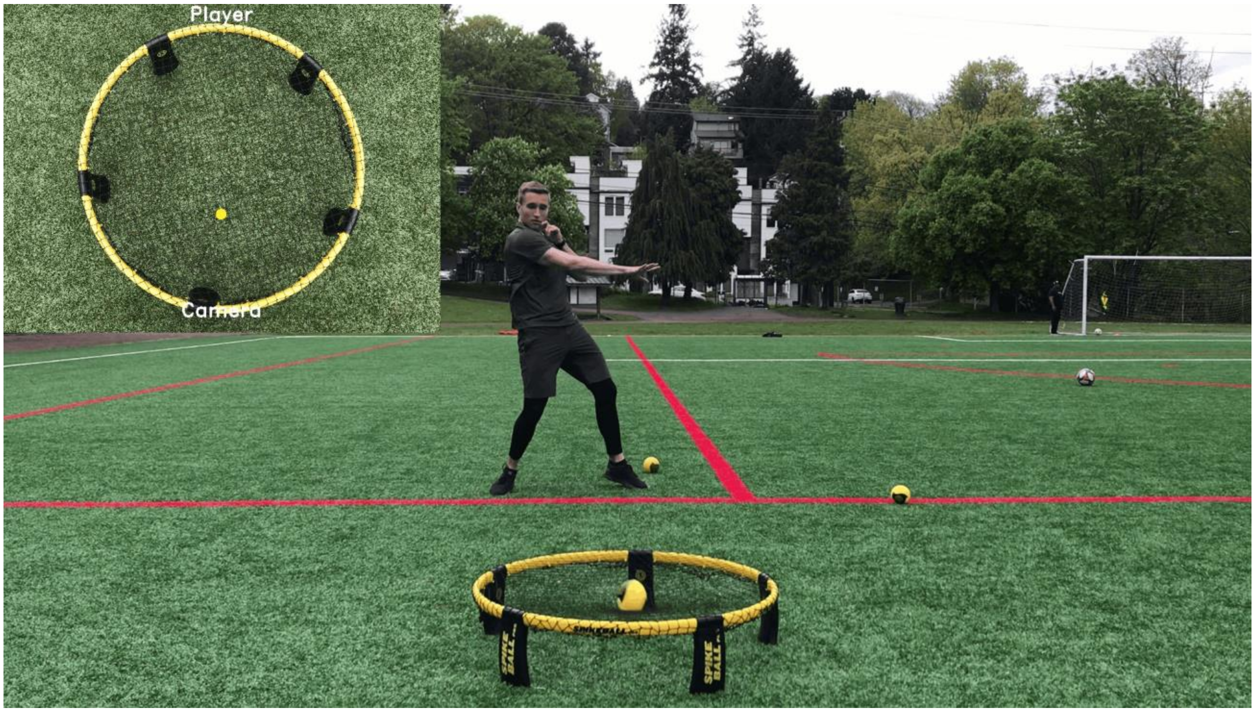
For my Capstone, I decided to continue off a previous personal project of tracking sports analytics in the sport of roundnet. Since a big part of the sport is serving, I decided to use what I've learned to create a proof-of-concept tool that can determine where a ball hits the net and translate that to a top-down view of the net. This inspired me because I play this sport competitively and a couple of years ago, I wanted to see if I could develop anything that would get me close to what some of the larger sports have. The computer vision concepts that I have used in this capstone are region-based image classification and homography.

Methods

To accomplish this, I used Facebook's Detectron2 Faster R-CNN model trained with a custom dataset to create the image classifications and I used OpenCV's homography module to create a homography matrix between various images. I chose Detectron2 because I already had the custom dataset and infrastructure setup to get and track the objects in a video. I used OpenCV because it is the standard computer vision library for much of computer vision and because it also happened to have the homography module.

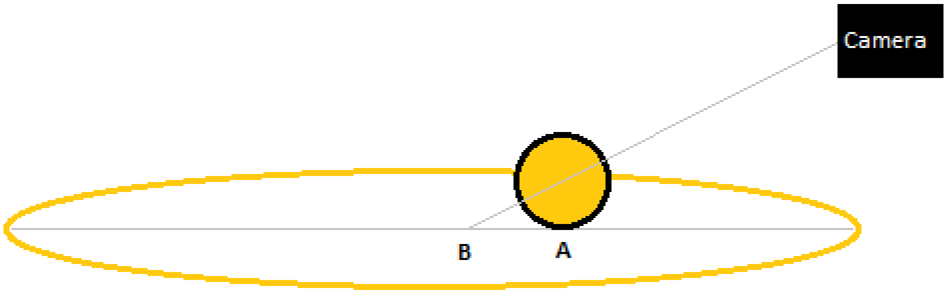
Since this was mainly a proof of concept, some of the data was gathered manually due to the time limitation for this capstone. While the image classification and homographies were done automatically, I hand picked the frames where the ball hit the net and manually gathered the points that could link the projections to one another.

Results



Within this looping gif, you can see within each frame that a ball has contacted the net. In the top left, the projected location of the ball can be seen along with the other ball hits that occurred previously.

Overall, I think the proof-of-concept was a success. The balls were mapped with relatively high accuracy which is what I was hoping for originally. There were two issues that I had that were somewhat related to one another. The first was that the net is not hard surface and so sometimes when the ball hits the net, the net caves which throws off the homography projection. The second was that since I was originally taking the center of the ball's bounding box as the ball's location, if a line were to intersect from the net through the ball to the camera, there would be some inaccuracy because of the unseen gap behind the ball as I've tried to illustrate in the image below.



A: Desired projection location
B: Actual projection location using center of bounding box.

Luckily, I was able to overcome this issue by using a combination of the center point and the radius of the ball added together in the Y direction.

Lessons Learned

Primary findings were that it was a success, although it was noticeably worse at projecting when the ball was near the rim of the net closest to the camera, that could be improved down the road. If I were to continue with this or maybe, do it again, I would probably perform key point tracking on the net itself instead of just bounding box object classification so that I could get the locations of the net's rim without having to manually find those points prior to projection mapping.