Project Name: SoccerNetPlus

AiJing Wu (awu53, 9081879174)
Michael Vanden Heuvel (mmvandenheuv, 9089226842)

Project Background and Motivation

We will address the problem of accurately detecting players on a soccer field in real time in videos. In addition, we will also estimate the localization of each player to create a top down 2D representation of the field with markers for each player.

Soccer is one of the world's most popular sports. As a result, the revenues associated with the sport are astronomical. Improvements to the way fans view soccer have the potential to increase revenues, affect the way billions of people enjoy the sport, and attract even more people to the sport. Utilizing a player detection and camera calibration model in unison could be used in scenarios such as auto-generating highlights, improving live graphics, and strategy analysis. Successfully recreating such a system would be rewarding both technically and personally because of our passion for soccer. We are passionate about this project because we are interested in the usefulness and range of applications of tracking people in video, enhancing user experience, and improving people's understanding of soccer.

Major Reference Paper:

Camera Calibration and Player Localization in SoccerNet-v2 and Investigation of Their Representations for Action Spotting

The Current State-of-the-art

1. OpenCV object tracking

   OpenCV is really popular and is extensively used in the field. It has a bunch of built-in functions to perform various object-tracking activities. If being used, we need to be careful with which function we're choosing to use, as each of them is distinctive with advantages and disadvantages.

   We are also aware that OpenCV will be the best practice for us to do a simple tracking system on our own, which will be our back-up plan if our current plan hits the wall so hard.

2. MATLAB tracking

   There are several functions in the Computer Vision Toolbox. The tutorials are well drafted, so we will be well-guided (see Tracking and Motion Estimation - MATLAB & Simulink). The major concern is about compatibility, as building an app needs to use other programming languages, which are not always able to collaborate with MATLAB.

3. MdNet

   The full name is Multi-Domain Convolutional Neural Network Tracker. It is built based on convolutional neural networks. Honestly, it looks to be a bit harder to

get our hands on (see https://arxiv.org/pdf/1510.07945.pdf). Also, it seems to be specialized at detecting some body movements, which could be useful in the further stage of our application, but still beyond the concern of this specific project.

4. DeepSort object tracking

   DeepSort uses YOLO v3 for computing the bounding boxes around the objects in the videos. This is a very cool algorithm with an identification model and estimated tracks (see https://arxiv.org/pdf/1703.07402.pdf). This might be in use for us if we want to split all players and find their individual paths.
   Also, this is the technique SoccerNet-v2 is using, so it will be our first touch on solving our proposed problem.

Goals on Reimplementation

We are planning on re-implementing an existing solution, with a better applicable scenario. We are glad to see that the existing model could calibrate the video tape to a 2D spotting localization representation, which will be very useful for fans to analyze the game and get more involved into understanding the team's strategy decisions. We notice it is hard for people with no machine learning background to take advantage of this great model, so our goals are:

1. Keep developing the existing model, and hopefully to get a more accurate spotting representation;
2. Design a web application to ease the process of using this model;
3. If time permits, we would propose some choices of general analyses based on the 2D representation generated.

Our contribution and creative aspects

1. Position analysis - examples: how good is the position of the players on each team (like analyzing a chess position)? Who is the best person to pass the ball to?
2. Simulation - map position of players from video frames into a 2D simulated environment.
3. Team separation - identify which players are on each team
4. User input - Improve inserting new data to analyze

Measurable Results

Quantitative results correspond to the qualitative questions:

- What is the success rate of identifying players within an image?
- What is the success rate of locating players within an image?
- How many seconds can the program track a player?
- What is the frame rate of the simulated field (can it keep up in real-time)?

Qualitative results will be multi-layered, including:

- Can our program identify player(s) within an image?
- Can our program identify where in the image the players are located?
- Can our program track players from frame to frame within a video?

- Can our program correctly map player locations in an image to player locations on a field?
- Can our program be interacted with effectively via some sort of user interface (command line, web app, etc)?

Estimated Timeline
(definitely subject to change as we navigate the project's progress):

Week Range	Tasks	Deadline
Oct. 2 - Oct. 8	Get familiar with existing technologies Decide the main technology to use Write proposal	Proposal (5%)
Oct. 9 - Oct. 15	Gather raw dataset Learn the main technology Split tasks within the group	
Oct. 16 - Oct. 22	Experiment with existing models and code repos to figure out how the software works	
Oct. 23 - Oct. 29	Train model Make adjustments to the existing model Started on 2D spotting	
Oct. 30 - Nov. 5	Train model Keep working on 2D spotting Start testing model	
Nov. 6 - Nov. 12	Hopefully the major machine learning part is done Keep testing model Write report	Midterm Report (5%)
Nov. 13 - Nov. 19	Web app UI design Fix compatibility issues	
Nov. 20 - Nov. 26	Web app UI design Integrate all parts together	
Nov. 27 - Dec. 3	Finish up all minor issues Design web page contents	
Dec. 4 - Dec. 15	Prepare final demo Finish webpage	Final Presentation (10%) Project Webpage (15%)