**FIZ 437E Project**

**Intro and Project Definition**

Football is one of the most popular sports in the world. The popularity of the game requires the use of many different methods to monitor and evaluate the performance of players. However, manually keeping track of player statistics is very difficult and there is a high likelihood of making errors. Therefore, image processing technologies have become an important tool for monitoring and evaluating player performance.

Image processing makes it much easier to keep track of player statistics. For example, Amazon's "Amazon Rekognition" product uses facial recognition and object recognition technologies to identify player identities and positions. As a result, player statistics can be automatically and accurately recorded.

When performing image processing, single cameras work better than multi-cameras. Single cameras allow us to achieve better results in areas such as player position changes. Tools like YOLO (You Only Look Once) and OpenCV can be used to track player positions and movements using single cameras. YOLO is a tool designed for real-time object detection. YOLO uses CNN to predict the location and classification of objects in real-time. OpenCV is a library used in image processing, computer vision and machine learning. With OpenCV, studies such as object tracking, face recognition, motion tracking and position tracking are performed.

Single cameras are better than moving cameras for several reasons. For example, single cameras can better monitor player positions and movements. Single cameras can better track player position changes because position changes are only monitored from one angle. Single cameras collect less data, so player movements are tracked faster and more accurately.

In conclusion, image processing technologies play an important role in monitoring and evaluating player performance. These technologies ensure that player statistics are recorded correctly and automatically. Additionally, these technologies are used in many areas such as improving football fields, fair play in games and even determining transfer policies for football teams. Among image processing technologies, YOLO and OpenCV are the most important tools. Single cameras achieve better results than moving cameras.

My aim in this project was to actively take the positions of the players from a football video and mark them on a 2d map. This way I would have extracted a map video from the player location coordinates data. With this map, I can make visualizations such as heatmaps, pass maps, shot maps by making inferences from player data in the later stages of the project.

The technologies I intend to use in the project are YOLO and OpenCv. For deep learning, I use tensorflow-keras.

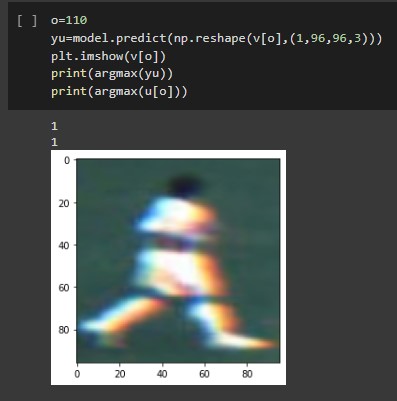
**Experiments and Code**

First of all, we stopped the match video and took screenshots and divided these images into 3 classes. We had 3 separate groups as A team, B team and referees.



Photos From Team A,B and Referee

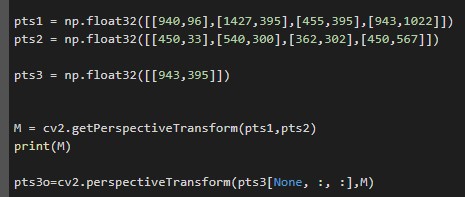
Then we separated these photos and their classes as X-Y data. With this data, we established a deep learning network. This deep learning model did not take long to set up and its prediction success was 100%. We also saved this deep learning model on the notebook. We will use this model on video data.



A photo of the model's estimation

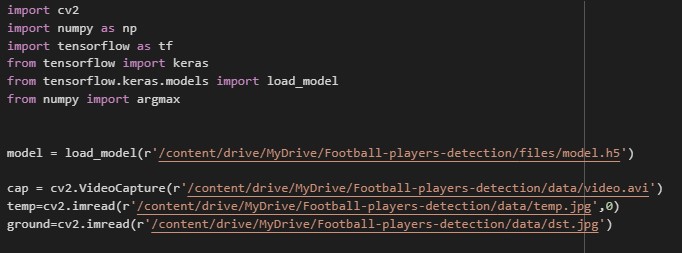
Afterwards, we took a screenshot from the match video we had. In our video, we used an easy method as the camera angle was never distorted.

For this method, we need a screenshot from our match video and an empty 2d football field photo. By marking the coordinates on the football field photo where the field is visible in our screenshot, we actually mark within which border the players we follow on the video will be in the photo. For this process, we use the cv2.getPerspectiveTransform method.



The code block where we do this operation

Finally, we come to the subject of capturing over video. I'll be explaining what I'm doing step by step.



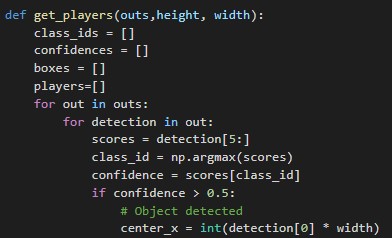
First of all, we add the libraries I need, the model we trained before, the ball we have and the 2d field photo and our video.

After that, I create 2 blank videos with the cv2.VideoWriter command to save the videos. I also add the yolo files with the cv2.dnn.readNet command.

Afterwards, we mark the object we captured with a function on the 2d map. What we do in this process is to mark the locations of the captured object on the 2d map. If it is the player we marked, we mark it with different colors depending on which team it belongs to. If it's a ball, we mark it with a different color. (We limit the number of balls to 1 because, as I explained in the class, when there are bald players on the field, it is perceived as 2 balls on the field :))

Then we write the function of catching the player over the video. This was honestly the part that challenged me the most. I was able to create a project by getting a lot of help from the sources on the internet and combining them. What we are doing here is what we caught in our video. Are the objects we caught actors? If so, from which team? Is it the ball? Or is it the referee? According to them, we mark the objects we capture with boxes on the video.

We set a confidence ratio of 0.5. The confidence ratio of 0.5 worked well in my model, I could have increased it further if it was making the wrong predictions.

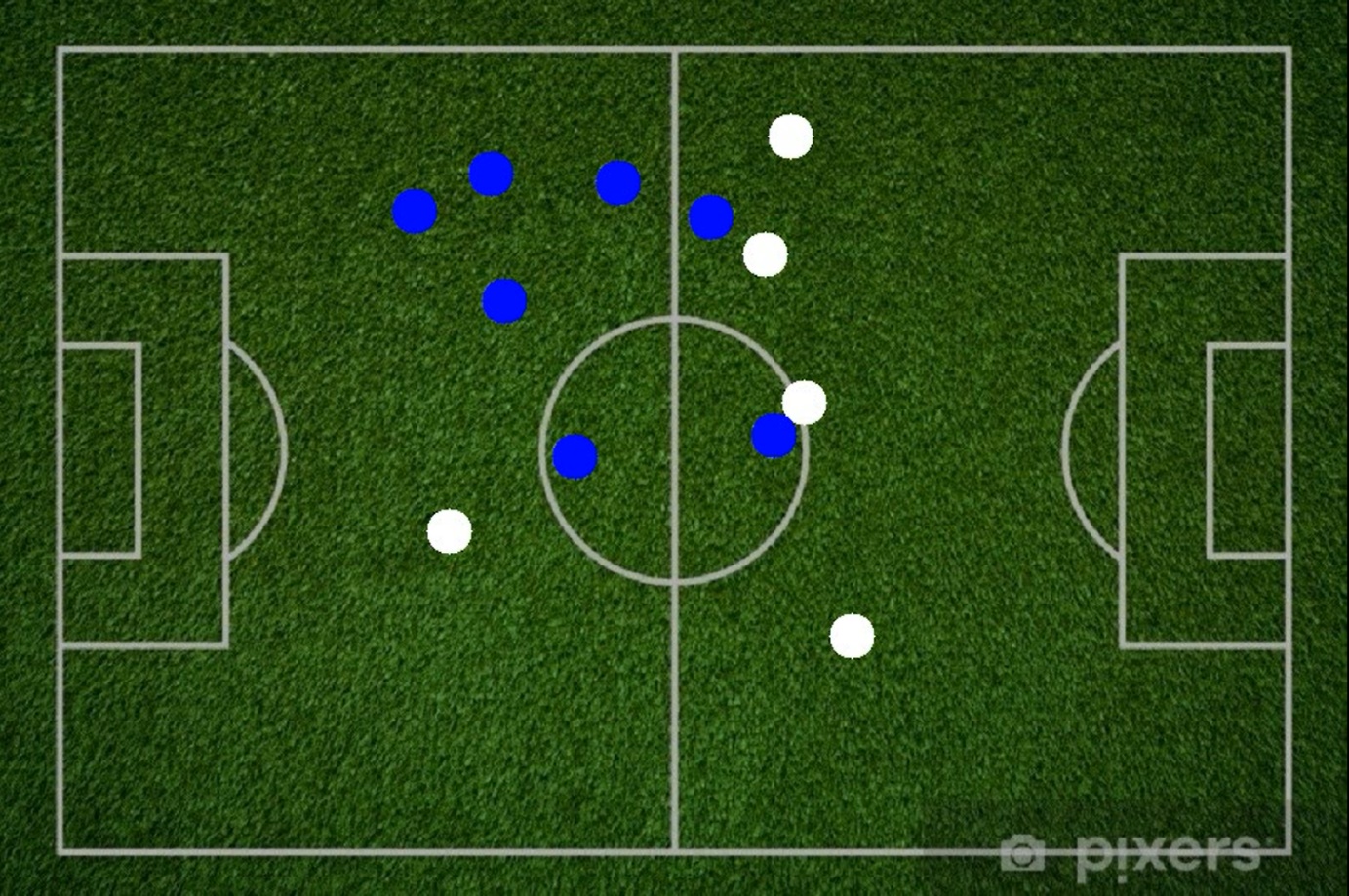


The input part of the function

Finally, we set up a loop that traverses our entire match video frame by frame, using the functions we wrote in these frames, and making captures with our model. We used the resize command because some frames looked bad. Afterwards, we made the functions and markings on this frame and added this frame to the blank video we created. With our other function, we made markings on the 2d map according to the locations of the markings, and we added the markings we made on the map to the 2nd blank video we created as a new photo.

This process took so long that my transactions via google colab took 3.5 hours. The total length of the created video is about 2.5 minutes.

As a result, we created 2 different videos. I leave screenshots from the videos below.



Screenshots from the videos we created

**Results and Future Works**

As a result, with this project, we captured the players in a match video shot with a single camera angle and marked them on the 2d map. Further in this project, visuals such as a heatmap from a single player's movements, or passing traffic from the ball's movements can be made.

The project can be developed and optimized so that it can work better and faster. It works very hard like this. While I was doing this project, I realized that if we are doing an image processing project, the resources we have in this project must be correct. We need a decent video source as data, and we need a powerful gpu. I believe we can do video processing much faster thanks to this gpu.

In the very advanced stages of this project, a huge project can be made, such as estimating what the player will do based on their movements and the positions of other players. But the data required for this project is so large that it may actually be one of the most comprehensive projects ever to be done.



Merged final video