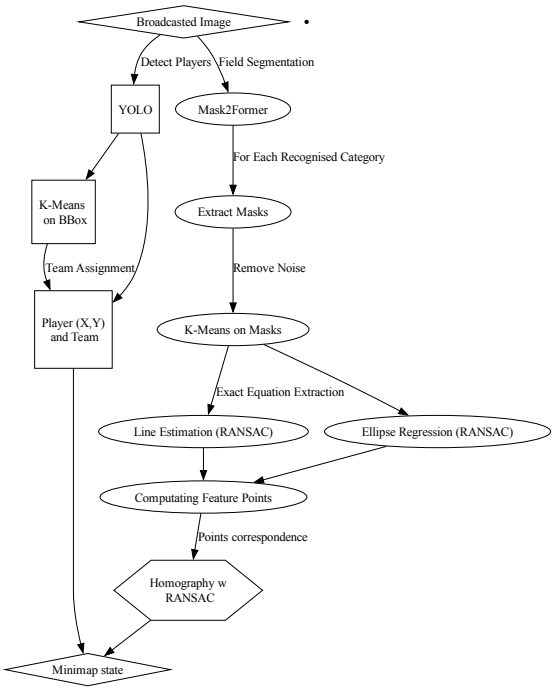
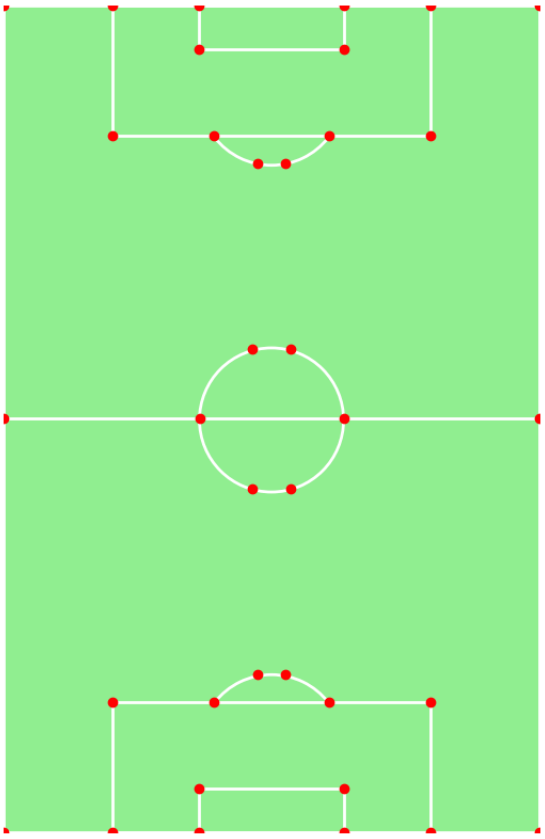
Computer vision project

Leonardo Bandera and Giacomo Bruno



(a) Conceptual map of project



(b) Fundamental points for H

1 Introduction and Overview

Our computer vision project aims to transform a broadcasted image of a football field into a minimap that displays players and their respective teams. As summarized in Figure a, our process involves two main pipelines. The first pipeline focuses on player detection and team assignment. The second, more complex pipeline, involves the automatic computation of homography, which is a non-trivial task. Initially, we had to localize various field markings, such as the penalty area, goal line, midfield line, midfield circle ecc, categorise them and extract exact equations. For the first task, we fine-tuned a segmentation transformer known as Mask2Former. Subsequently, from the segmentation masking we extracted the equations of these entities employing both Ransac with ellipse and line fitting, and K-means to remove the noise of the segmentation (extract only the white part). Once found the analytical equation we proceed with computing feature points that are seen in figure b. For our homography, we computed a set of 36 notable points within the minimap coordinates. These points serve as potential correspondences used in our homography computation and to ensure the most accurate homography, we again employed RANSAC. This notebook will be as a guide for the presentation of the project

2 Homography Computation for a Football Field

Since our project is involved and complex we present you first our idea to build an homography and later how we arrived to it. To compute the homography of a football field, we explored several methods, including the use of conics,

$$C' = H^{-T} C H^{-1}$$

lines,

$$l' = H^{-T} l$$

points and combinations of these features, however each one of those had some difficulties either in the photo(lack of features) or in the effective computation of the constraints. Also our objective was to automate the process for rectifying the image without human intervention, so we decided to focus solely on point correspondences. Specifically, we established a set of 36 points on the MiniMap, illustrated in Figure 2. This set includes 21 line-to-line intersections, 6 circle-line intersections, and 9 points where a tangent line passes through a point on a circle. This approach enabled us to automate the homography computation process effectively, ensuring that we have a sufficient number of points for the computation under various photographic conditions. Even in scenarios where only a circle and a line are visible, our method can generalize to compute the homography.

By focusing solely on point correspondences, we streamline and robustly generalize the homography computation across different conditions and views of the football field. This method not only simplifies the automation but also enhances the reliability and flexibility of the system in diverse scenarios. Once we decided that this was the solution we needed to compute the notable points coordinate in our photo in order to have the correspondences.

3 How to get Points

Segmentation

3.0.1 Challenges

When given a photo of a football field, our initial approach was to use the Canny edge detector coupled with Hough transform, aiming to capture the structural lines of the football field. However, we encountered two major issues. The outputs of these varied depending on the photo, the pitch, as many fields have different grass patterns and different lighting. In order to address this we had to change the hyperparameters of the Canny edge detector and the Hough transform, specifically the sigma for smoothing, the line length in the Hough transform, and the thresholds depending on the situations. In a nutshell, it did not generalize well, not even in a single photo since perspective changes affect the appearance of sidelines, not to mention the occlusion caused by players covering part of the lines. Consequently, we researched and found a dataset used for camera calibration, which includes photos and coordinates of some key points, e.g., the goal, type of lines. We adapted this dataset for semantic segmentation, creating masks for lines, arcs and goal and trained a Mask2Former model that showed promising results. This subsection explains the dataset and the model, including our training approach.

3.0.2 Model

Our project utilized the Soccernet dataset, a rich collection of annotated images tailored for football computer vision challenges. The dataset comprises 16k JPEG images and JSON files

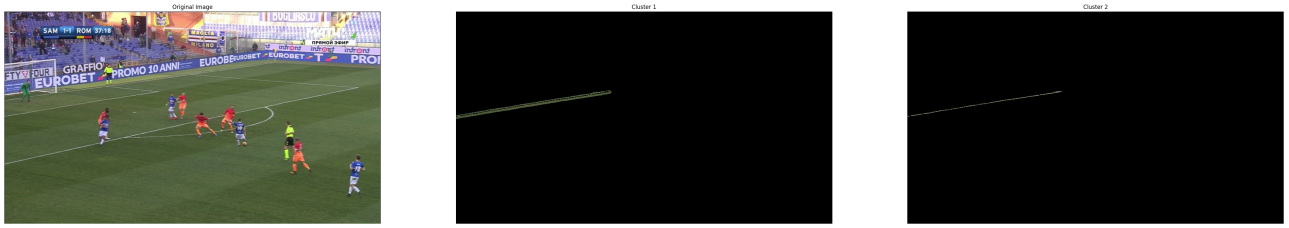
detailing the coordinates of various elements seen in football broadcasted image, such as goal-posts and pitch lines and even cateogories of line . Each image comes from multiple camera angles, capturing all possible situation of a football field. We modified this dataset to train our segmentation models and published it on Huggingface.

Once we found the dataset we looked on the web for state of the art segmentation network and we found Mask2Former, a universal architecture renowned for its ability to tackle semantic, instance, and panoptic segmentation with a single model. We decided to give it a try because we were really intrigued and indeed even if faced with computational resources difficulties we obtained great results, obtaining a mean IoU of 0.69 on the test set. To prepare our dataset for training, we converted the images and annotations into a png with the masking. We applied a series of transformations to normalize and resize the images, specifically tuning the input for optimal processing by the model. We encountered a lot of break in training due to Colab's inactivity timeouts so we saved the model at the end of every epoch. Additionally, we experimented with the MaskFormer model, which yielded a mean IoU of 0.63. Although slightly less effective, this model still provided valuable insights into segmentation performance. Given the constraints of our academic schedule and the extensive training times required, we thought of training a conventional CNN, a topic extensively covered in our coursework.In the coming days, if time allows we plan to explore CNN-based segmentation to compare. Here you can see some results of the mask2former

(a) Result on the left the photo and on the right the masking

Equation Computation

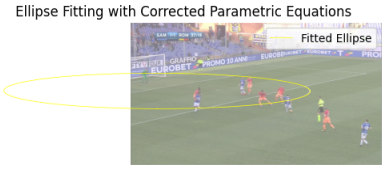
Upon analyzing the results from Mask2Former, we observed that masking, as expected, was not completely perfect; noise was apparent, particularly when players overlaid it or if the line was distant, with the green of the pitch heavily present in the masked pixels. Given the importance of accurately representing lines or circles for our homography, it was crucial to have a robust algorithm. We first performed K-means clustering with two clusters on the masked pixels of key lines, like the midfield or the main penalty line. After identifying the two centroids, the cluster whose centroid was closest to the point $(255, 255, 255)$—representing pure white—was designated as the true mask. However, noise was still present, possibly from players wearing white shirts over the lines or in general covering the lines. To address this, we applied the RANSAC algorithm to fit a robust line to the pixel coordinates. The same procedure was replicated for the three conics present on the field, using an ellipse-fitting RANSAC on the clustered mask. The results were very satisfactory, yielding the precise equations of the football field lines in the photo. As you can see in the figure below.



(a) Example of clustering

(b) Line with ransac

(c) Line with ransac



(d) Ellipse with ransac

3.1 Point Mapping

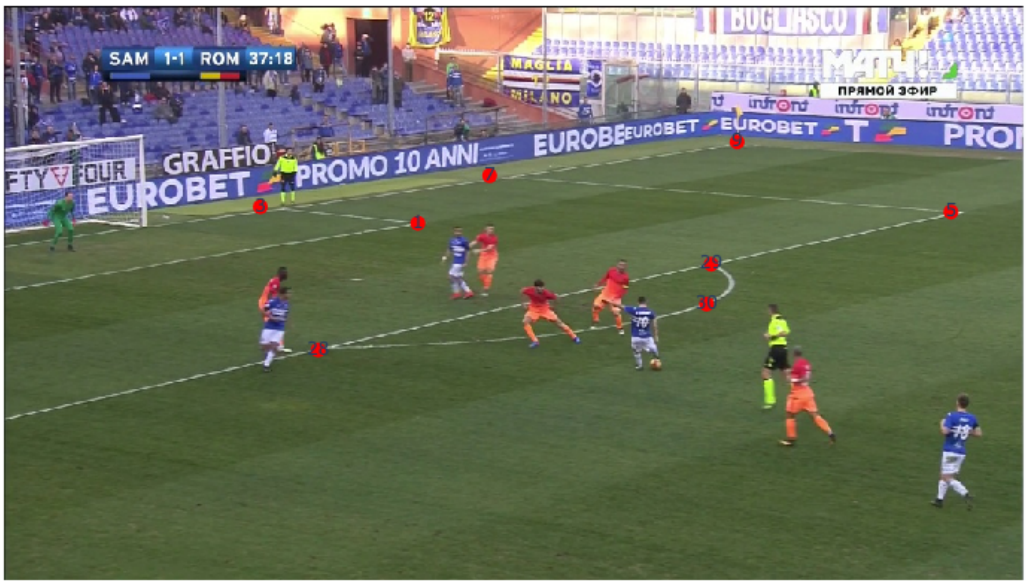


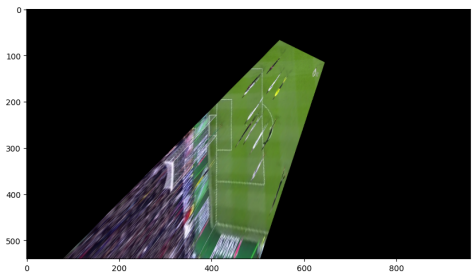
Figure 4: Point mapping

Once obtained the equations it was just a matter of smart bookeeping in order to compute the right intersection so we built a dictionary that for every line- circle key it had as values the index of the line to build the intersection or the point for tangency.So given a line in homogeneous coordinates $[p, q, r]$ and an ellipse defined by $ax^2 + bxy + cy^2 + dx + ey + f = 0$, the intersection points are calculated by converting the line to a slope-intercept form if it is not vertical, substituting this into the ellipse equation to derive a quadratic in x or y, and then solving this quadratic to identify intersection points.

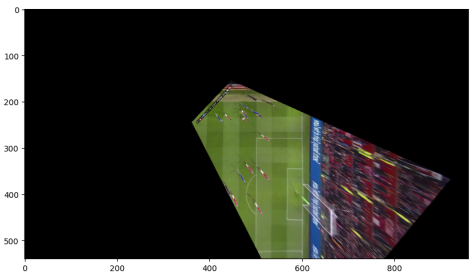
Through these methods, we are able to compile a dictionary that ultimately maps each conic section to its relevant geometric figures, accurately capturing intersections and tangency points with 36 unique keys.

4 Homography Final

In our approach to computing the homography matrix H between two sets of corresponding points from different images, we utilize the Direct Linear Transform (DLT) method augmented with RANSAC (Random Sample Consensus) to handle inaccuracies and outliers in point correspondences. Initially, for each iteration within the RANSAC loop, four point pairs are randomly selected to construct matrix A of constraints, which is then subjected to SVD to find the optimal homography matrix H. Then H is validated by calculating the reprojection error for each point pair, determining inliers as those with errors below a predefined threshold. By iterating this process n times, we aim to maximize the number of inliers, thus refining H to a robust estimation that accounts for potentially erroneous or outlier point correspondences.



(a) Results



(b) Results

5 Introduction and Overview

6 Player Detection Pipeline

In order to develop an effective MiniMap, we needed precise player positioning. For this purpose, we experimented with multiple YOLO models, fine-tuned with 315 annotated images from football broadcasts with different configurations and hyperparameters. Among these, YOLOv8x, yielded the best overall performance on our dataset. This specialized tuning was essential as the generic YOLO model failed to differentiate between players and non-player figures like managers. Our modified model performed well, achieving actually impressive metrics scores, as shown below. In the realm of lower resolution images, such as those incorporated in our dataset for the segmentation , some challenges arise, notably in distinguishing between players and referees, and occasionally experiencing difficulties in player detection over noisy background. To address these challenges, various strategies could be adopted. Implementing data augmentation techniques, expanding the dataset by annotating additional images, potentially with lower resolutions, or even transitioning to video data could be beneficial: leveraging video data offers the advantage of utilizing tracking mechanisms, aiding in the accurate detection and recognition of players within dynamic game scenarios.

7 Team Assignment

After determining players' bounding boxes, we analyze the cluster of the upper part of each player to assign team affiliations. In our player recognition system, we've implemented a TeamAssigner module to assign teams to detected players. This module utilizes color analysis techniques to distinguish players' team affiliations. The process begins with analyzing the colors present in the top half of each player's bounding box. This is achieved through K-means clustering, which helps identify dominant color clusters associated with each player, detecting

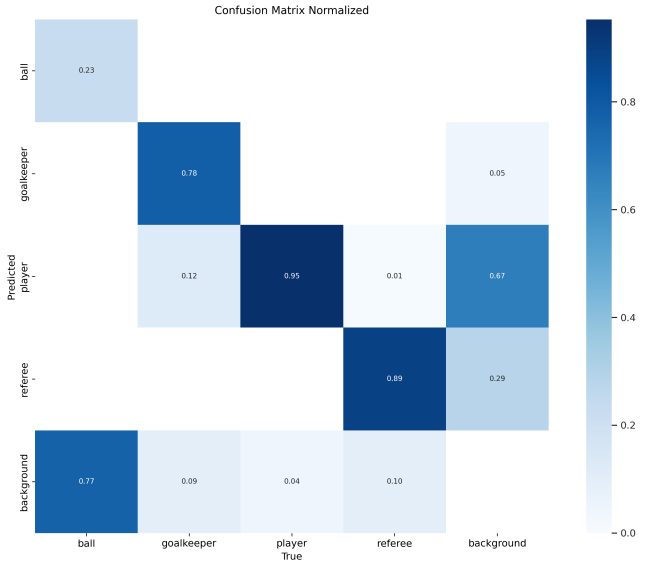


Figure 6: Fine-tuned YOLOv8x evaluation

a player color and a background color. Once the dominant color clusters are identified for each player, they are used to determine the team affiliation of each player. By clustering these colors through K-means again, the system effectively divides players into distinct teams based on their color similarities.

Figure 7: Results of YOLO

8 Final Remarks

Our project was divided into two sub-projects. The first, involving YOLO, was relatively straightforward and primarily served as a practical introduction to the necessary techniques for integrating player tracking into the MiniMap. The second sub-project was more complex, involving the recognition of field lines, deriving their equations, and developing an automated model for homography. A significant amount of time was dedicated to managing a dictionary of 36 reference points and determining which equations corresponded to particular geometric relations, like intersections or tangencies. This project allowed us to apply numerous concepts covered in class, leading to results that were quite satisfactory.

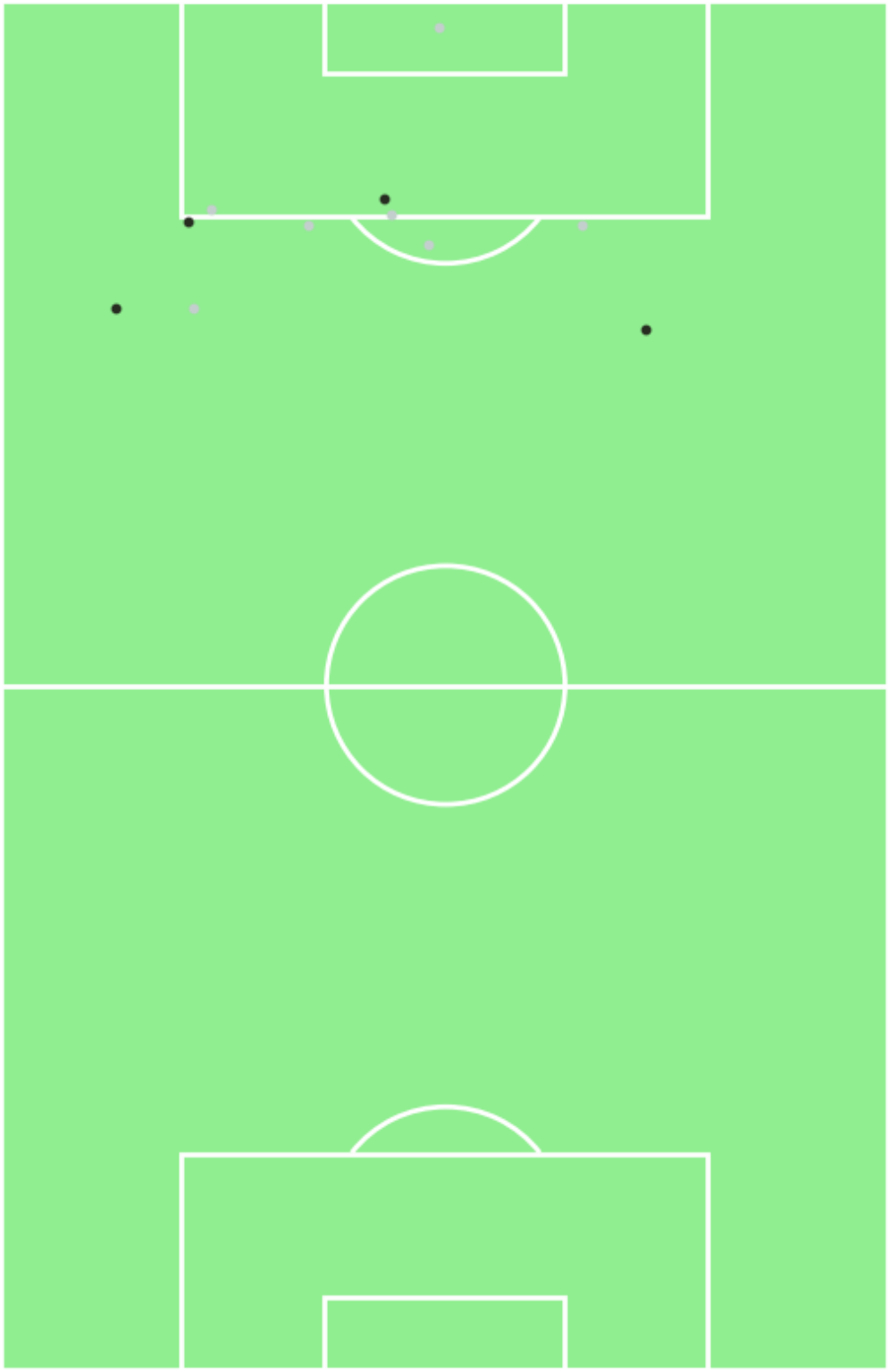


Figure 8: Final results