MARIO: Modular and Extensible Architecture for Computing Visual Statistics in Robocup SPL

Domenico Bloisi, Andrea Pennisi, Cristian Zampino, Flavio Biancospino,
Francesco Laus, Gianluca Di Stefano, Michele Brienza, and Rocchina Romano

https://github.com/unibasteam/MARIO-Modular-and-Extensible-Architecture-for-Computing-Visual-Statistics-in-Robocup-SPL https://sites.google.com/unibas.it/wolves/robocup/robocup-2022?authuser=0

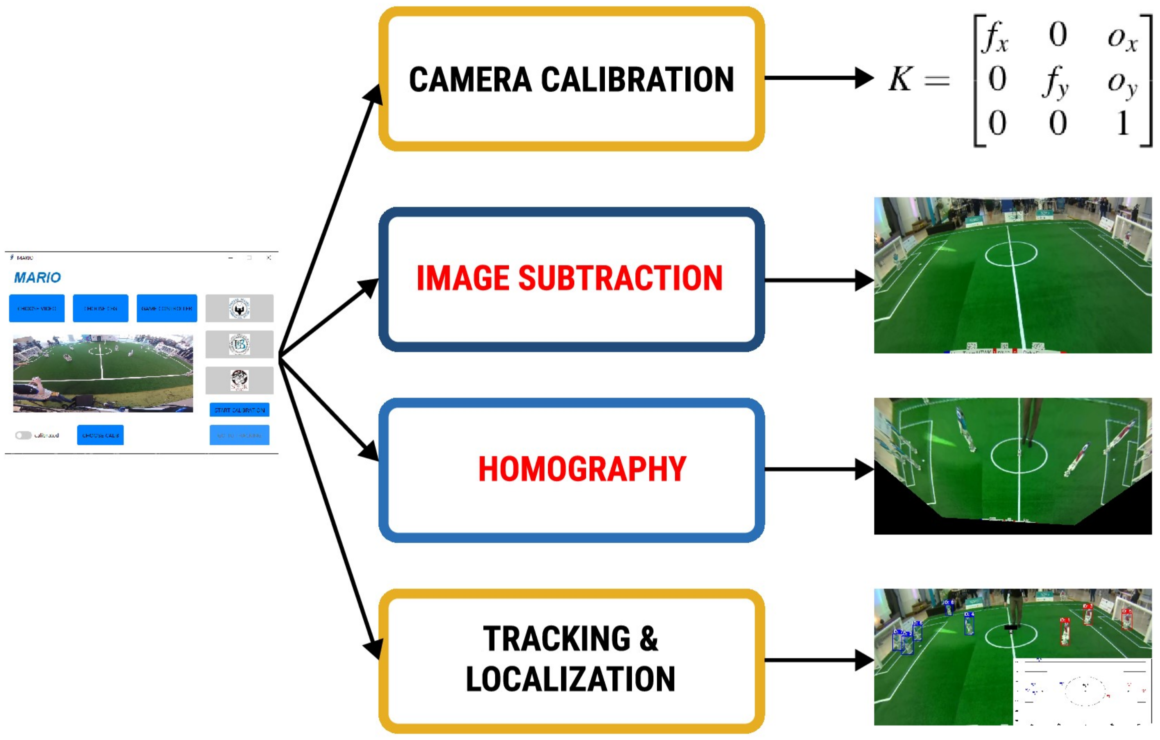
SHORT TERM GOALS

A preliminary camera calibration is performed to remove the camera lens distorsion.

An image subtraction tecnique combined with a NN model for field's keypoints retrieving is applied.

Homography is used to compute a plan view of the field. All process is performed in an automatic way.

A combination of YOLOv5 and StrongSORT models are used to track and localize the players and the ball. Informations are then reprojected on to a 2D field view.



CAMERA CALIBRATION

$$K = \begin{bmatrix} f_x & 0 & o_x \\ 0 & f_y & o_y \\ 0 & 0 & 1 \end{bmatrix}$$

IMAGE SUBTRACTION

HOMOGRAPHY

TRACKING & LOCALIZATION

Among a set of intrinsic parameters of the camera, those with a minimum reprojection error have been chosen.

Subtracted image and retrieved keypoints are used to infer the field's version.

Computed reprojection error helps in camera calibration operations and players identification.

Homography data and tracking data helps in the process of jersey number recognition and team assignation.

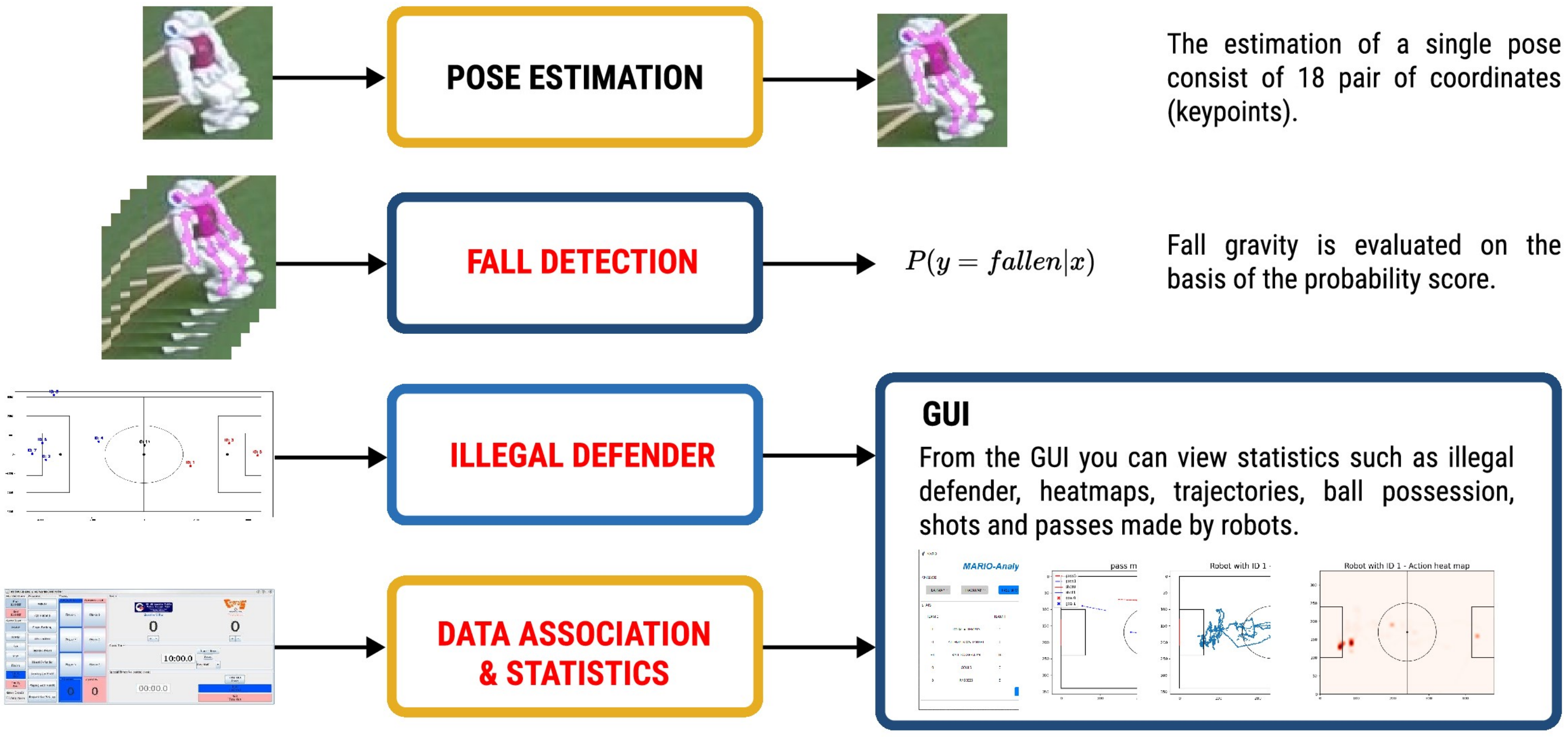
LONG TERM GOALS

A CNN network is used to perform the pose estimation of the robot. A custom dataset is created specifically for this task.

Based on skeletal information obtained, a ST-GCN model is used to perform the fall detection.

The tracking results are used to check if no more than three players from the same team are in the same penalty area. Their information, with GC data, are stored in a .csv file used for statistics.

Game data containg player information are stored in a .csv file. From the .csv file we have calculated we can extract the statistics about the game.



POSE ESTIMATION

FALL DETECTION

$P(y = fallen|x)$

ILLEGAL DEFENDER

DATA ASSOCIATION & STATISTICS

GUI

From the GUI you can view statistics such as illegal defender, heatmaps, trajectories, ball possession, shots and passes made by robots.

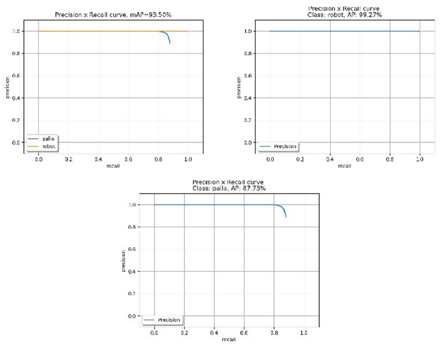
The estimation of a single pose consist of 18 pair of coordinates (keypoints).

Fall gravity is evaluated on the basis of the probability score.

HW REQUIREMENTS & RUNTIME PERFORMANCE

- **LOW-END HW:**
  CPU: Intel i3-6006U**, GPU:** Intel Graphics HD 520**, RAM:** 8 GB
  AVG FPS WITH POSE ESTIMATION: 3
  AVG FPS WITHOUT POSE ESTIMATION: 7
- **MIDDLE-END HW:**
  CPU: Ryzen 7 5700U**, GPU:** RTX 3050 Ti Mobile 4 GB vRAM**, RAM:** 16 GB
  AVG FPS WITH POSE ESTIMATION: 6
  AVG FPS WITHOUT POSE ESTIMATION: 16
- **HIGH-END HW:**
  CPU: Intel Xeon W-2135**, GPU:** Nvidia Quadro P4000 8 GB vRAM**, RAM:** 32 GB
  AVG FPS WITH POSE ESTIMATION: 10
  AVG FPS WITHOUT POSE ESTIMATION: 20

METRICS



35k images have been used as data. They where divided in three parts: train (24k), validation (3k) and test (8k). Data are labelled in 2 classes: robot and ball.

To improve execution time, we used YOLOv5 model for training the NN. The use of YOLOv5 for detection allow an inference in real-time. Also on CPU usage.

NOVELTY

Novel approaches proposed are highlighted in red.

INFO

For further information about the project and to download the Docker image with the source code, scan the QR Code.