ROMECUP 2024

INTELLIGENZA ARTIFICIALE E ROBOTICA

PER LA SFIDA DELLA SOSTENIBILITÀ OLISTICA

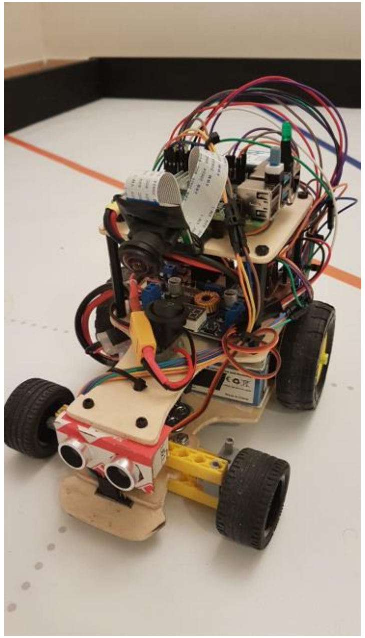
CONTEST CREATIVI DI ROBOTICA

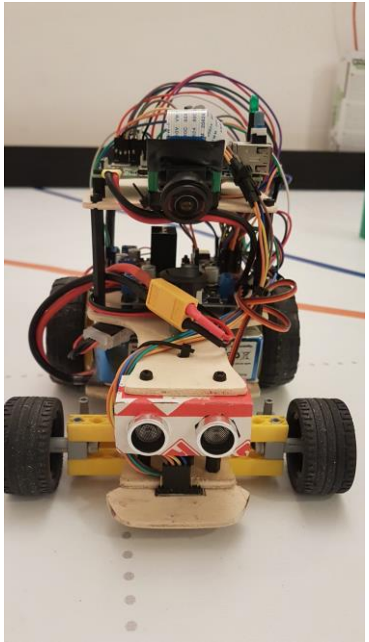
21 marzo, Università degli studi di Roma Tor Vergata

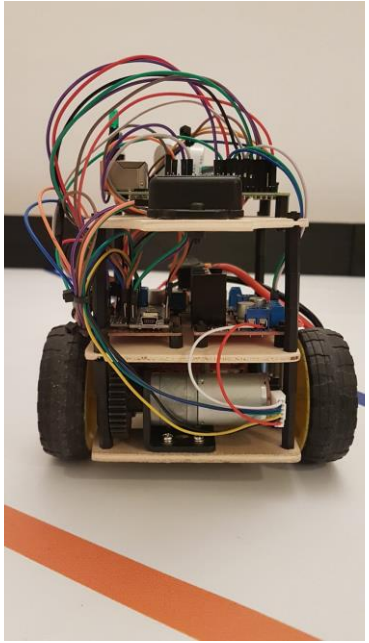
Volterrabots, DroneBOT

L.S.S. Vito Volterra -in collaborazione con– Tor Vergata

We have **crafted** a **custom** vehicle using the *Raspberry Pi 4 model B* as its foundation.







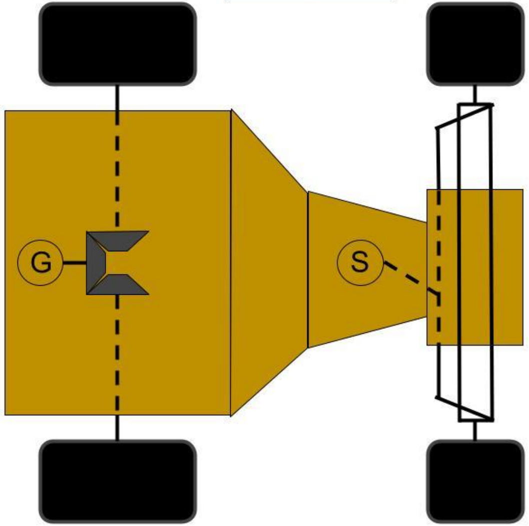
Mechanical schemes

View from the bottom of the vehicle.

The dotted lines are to make it clear that the mechanical piece is on the other side of the platform.

G = gearmotor

S = servo motor

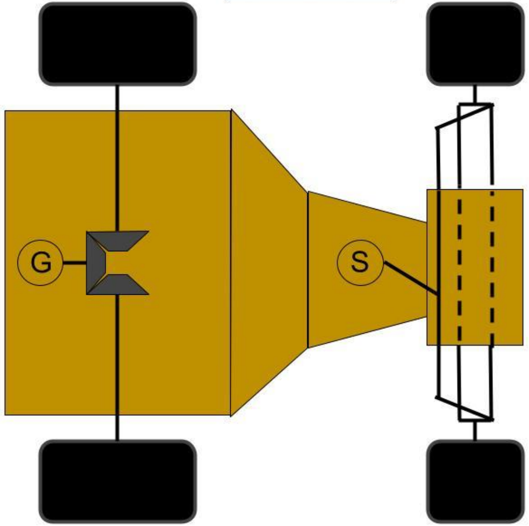


View from the top of the vehicle.

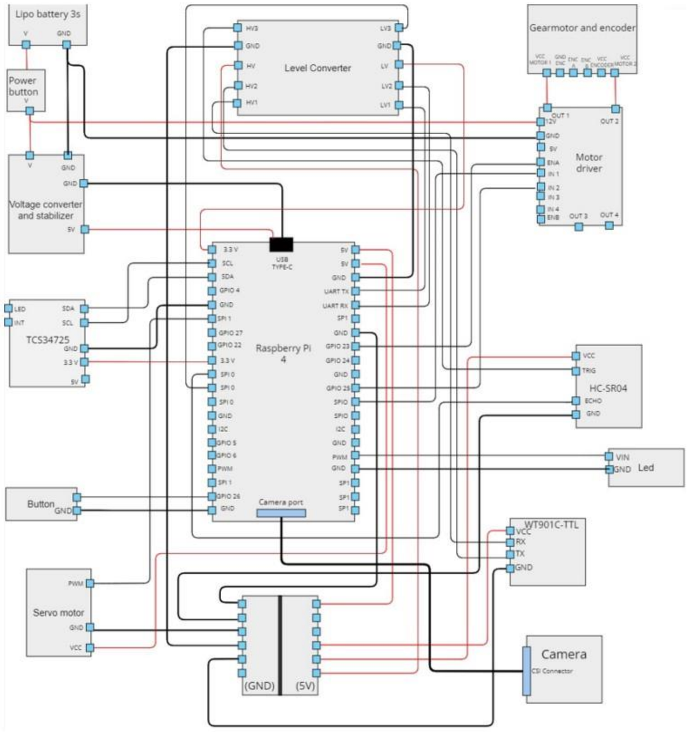
The dotted lines are to make it clear that the mechanical piece is on the other side of the platform.

G = gearmotor

S = servo motor



Electrical scheme



Here is a list of the hardware components utilized by the rover in the competition:

- *Gearmotor*

- *Tension regulator and stabilizer*

- *Servo motor*

- *Lipo battery 3S 1800mAh*

Overview of Control software - Key Python libraries

- numpy
- opencv
- PyTorch
- ultralytics
- websockets
- gpiozero

Overview of Control software - project architecture

Computer-rover communication

We utilize *websockets* to establish a TCP communication between the computer and the rover, swiftly transmitting the movement with low-latency. Upon successfully receiving the message, the rover employes the *gpiozero* library to produce a physical output via the actuators.

Multi-threaded parallelism

To enhance the performance of our programme, we leverage a combination of multithreaded and asynchronous routines.

Overview of Control software - project architecture

NVIDIA
CUDA

Performance for DL related task

To accelerate the inference process of our DL model, we load it onto the GPU (NVIDIA RTX 2060), benefiting of CUDA's resources optimization.
The results are impressive, with inference time on GPU being less than a tenth of that on the CPU.

How we capture the video source

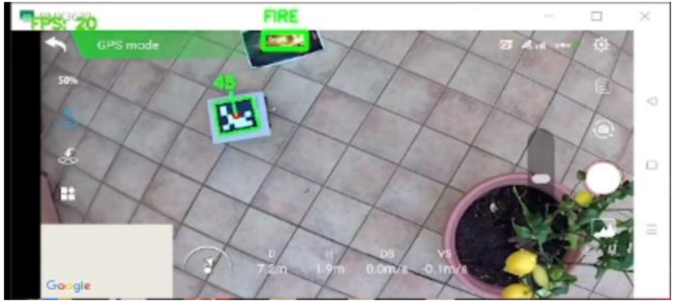
scrcpy

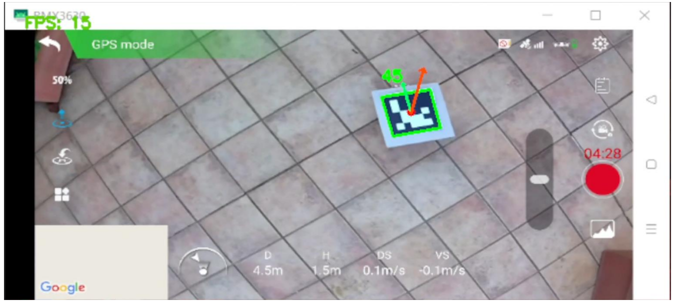
An **Open Source** application that mirrors Android devices (video and audio) connected via USB or over TCP/IP, and allows to control the device with the keyboard and the mouse of the computer.

- REPO: https://github.com/Genymobile/scrcpy

How we capture the video source

After having mirrored the image on the desktop, we generate a video source by repeatedly capturing its screenshots, preparing the image for further processing. Namely Aruco Marker recognition and Fire recognition.

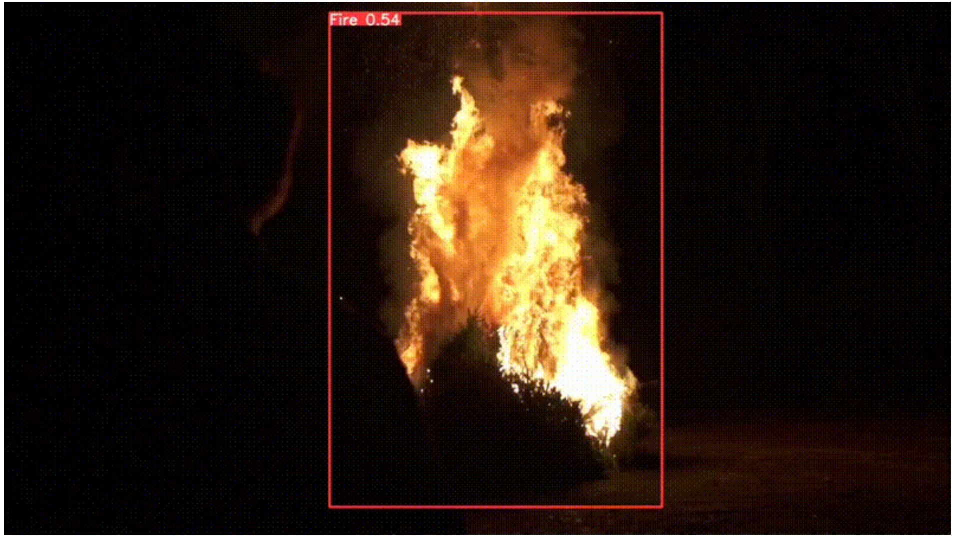




Two different examples of processed screen capture

Fire detection using an AI model

The model uses a custom implementation of YOLO v8

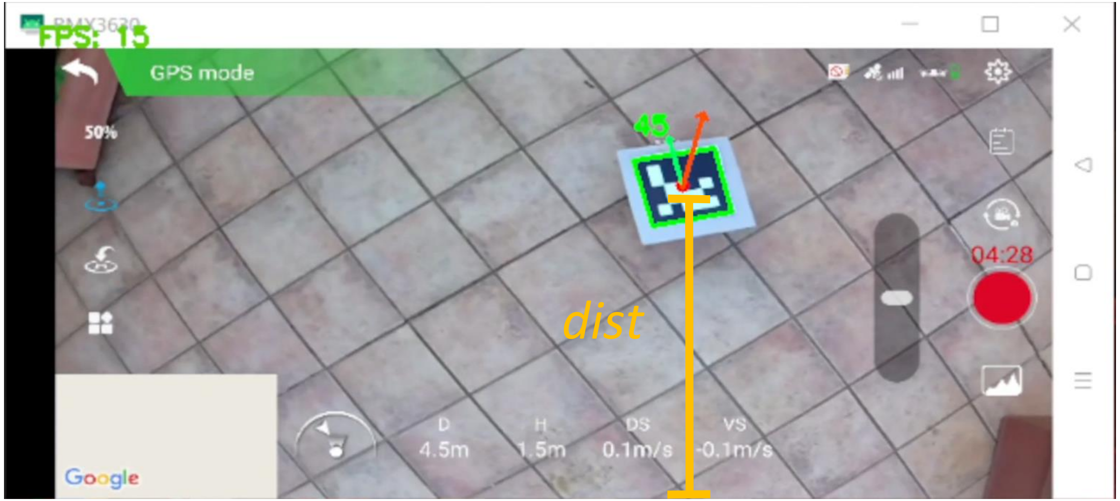


- TUTORIAL REPO: https://github.com/MuhammadMoinFaisal/FireDetectionYOLOv8

Something about the navigation

- The steering angle is determined by assessing the parallelism between the roll axis of the drone and the rover.

- Motor speed is calculated by considering the rover's position on the frame relative to a designated *sweetspot*, with pixel coordinates converted to meters using proportional correlations.



$$\mathit{dist} = \mathit{H\_ARUCO} * (\mathit{FRAME\_HEIGHT} - y - h/2) / h$$