MINISTRY OF EDUCATION

TECHNICAL UNIVERSITY
OF CLUJ-NAPOCA, ROMANIA

Real-time crowd identification and monitoring using UAVs

Student: **Andrei Rusu**

Coordinator: **Prof. Dr. Radu Dănescu**

Problem

- Crowds in public places = large health hazard
- Difficult to enforce social distancing
- Traditional methods are slow and require offsite processing

Objectives

- Implement CV method for estimating overcrowding & social distancing
- Leverage edge computing to enable real-time processing
- Use UAVs for fast deployment anywhere

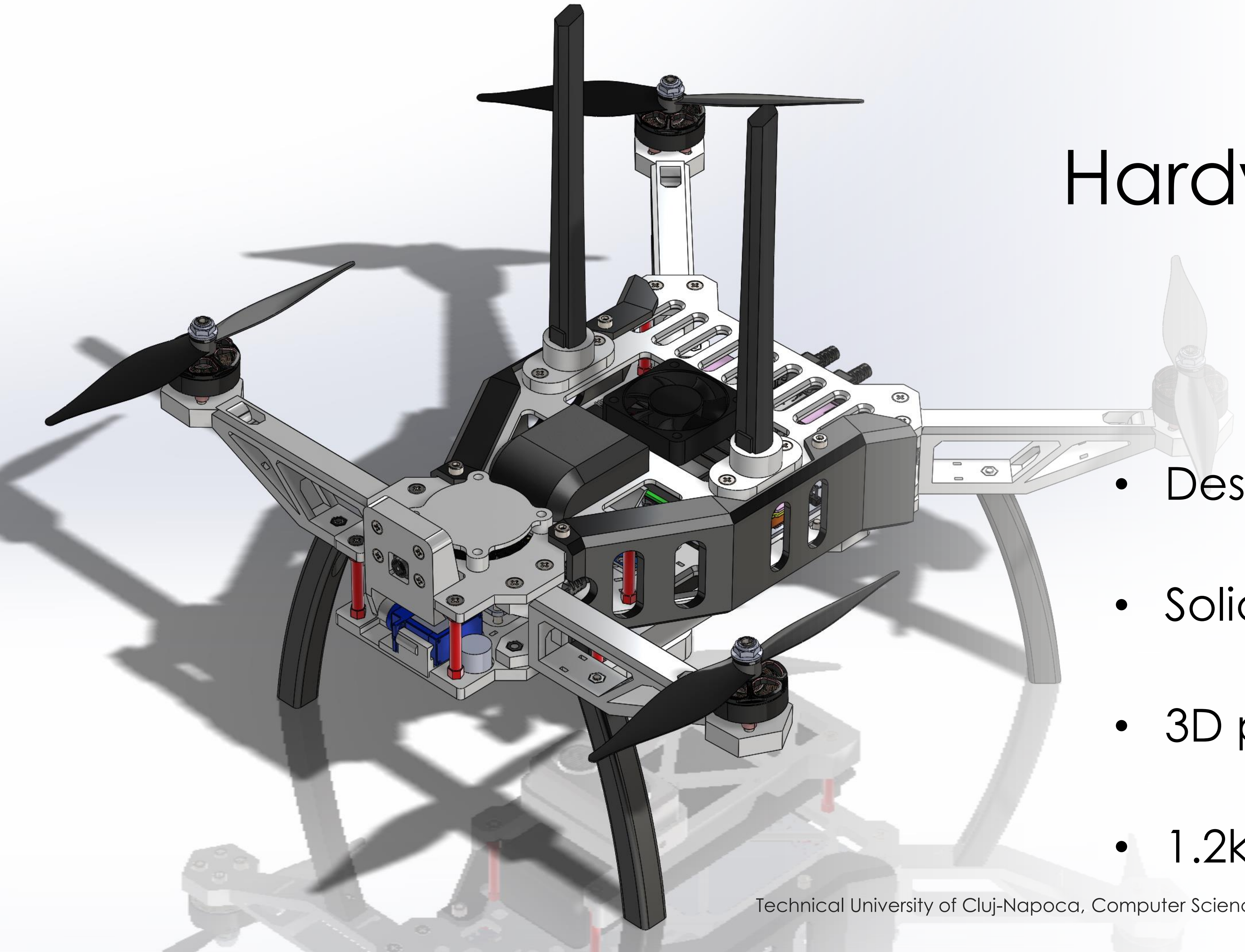
Proposed solution

Hardware

- Custom UAV chassis
- Main computer
- Sensors

Software

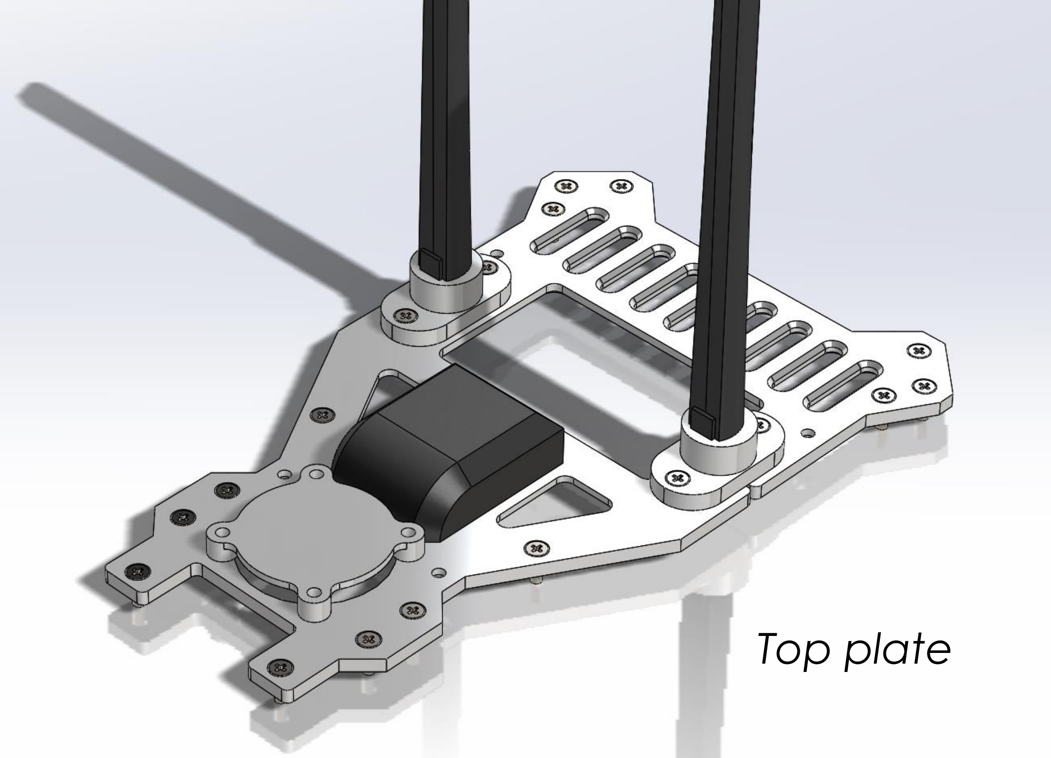
- Real-time detection
- Data Logging
- UAV control



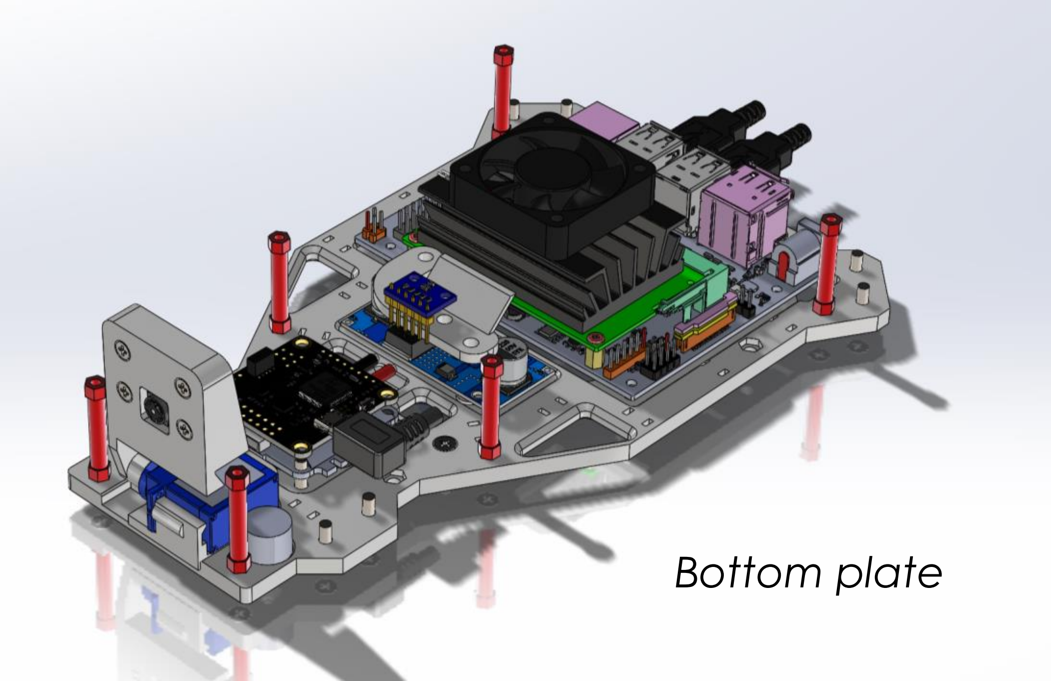
Hardware platform

- Designed & built myself
- SolidWorks CAD tool
- 3D printed with PETG
- 1.2kg, ~20min flight time

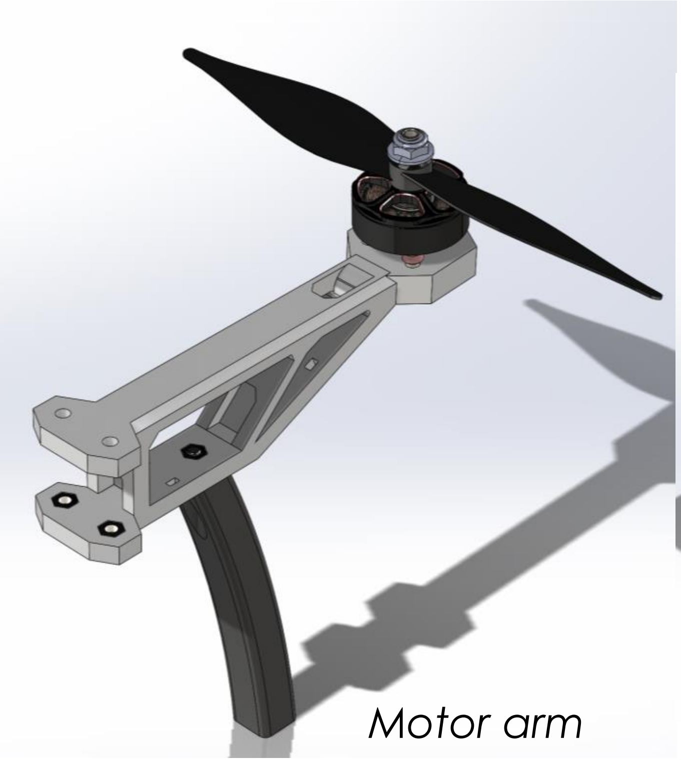
Designed assemblies



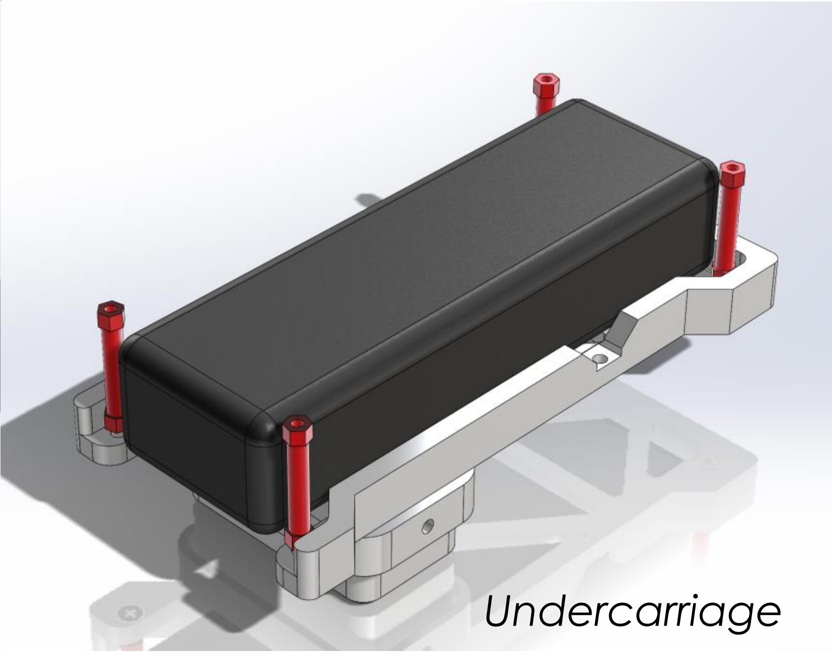
Top plate



Bottom plate

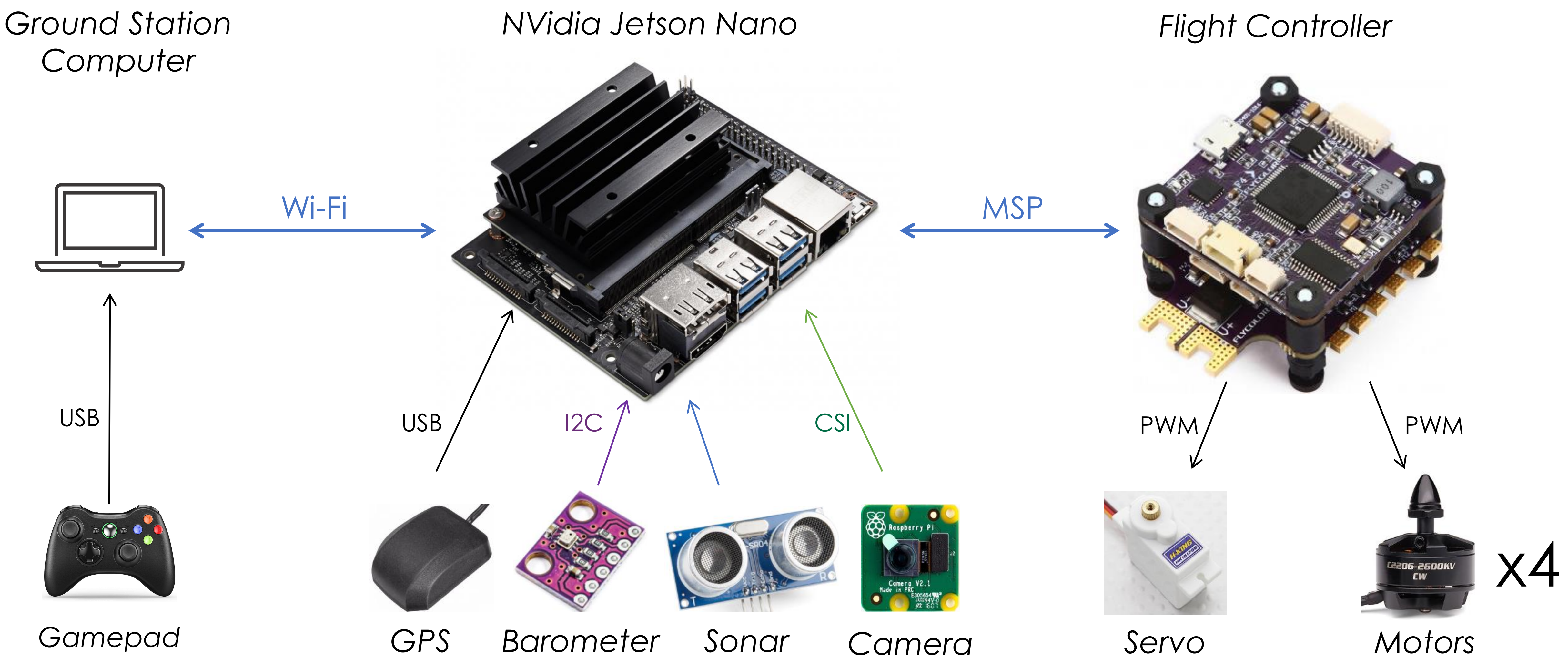


Motor arm

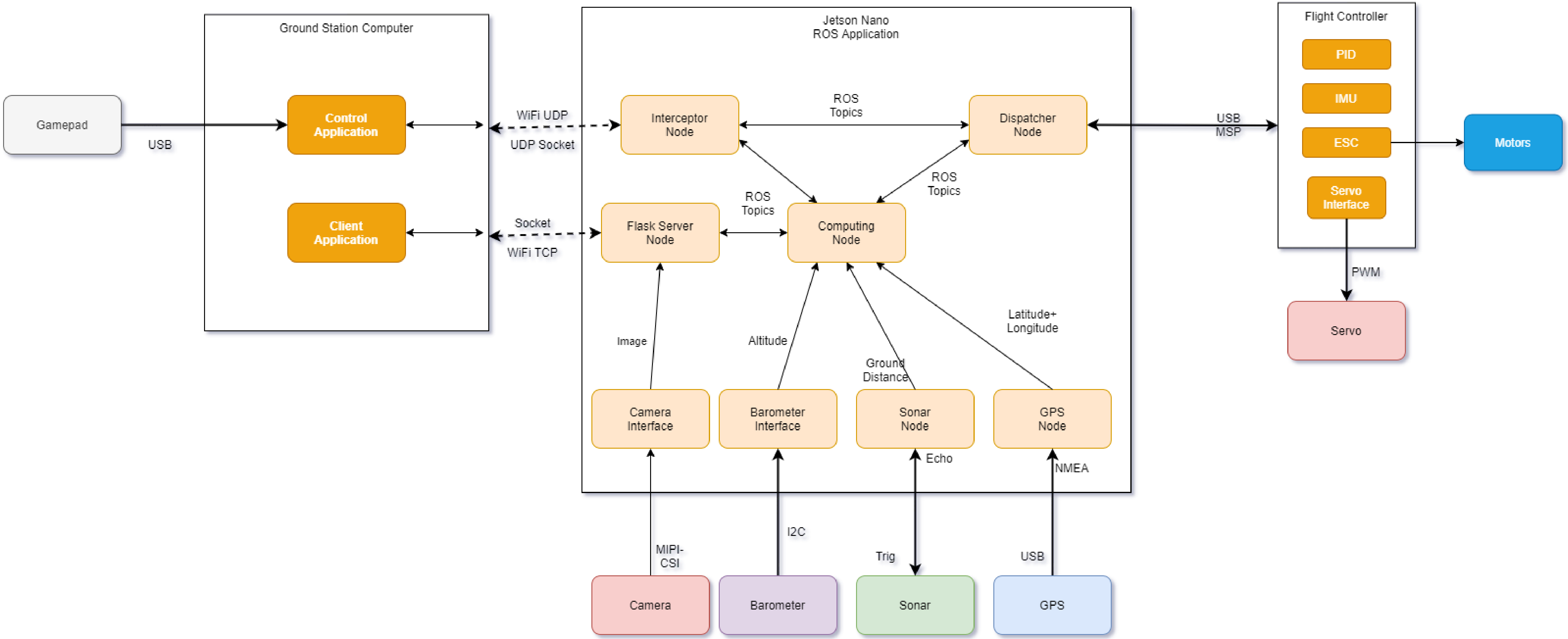


Undercarriage

Hardware components



Software components

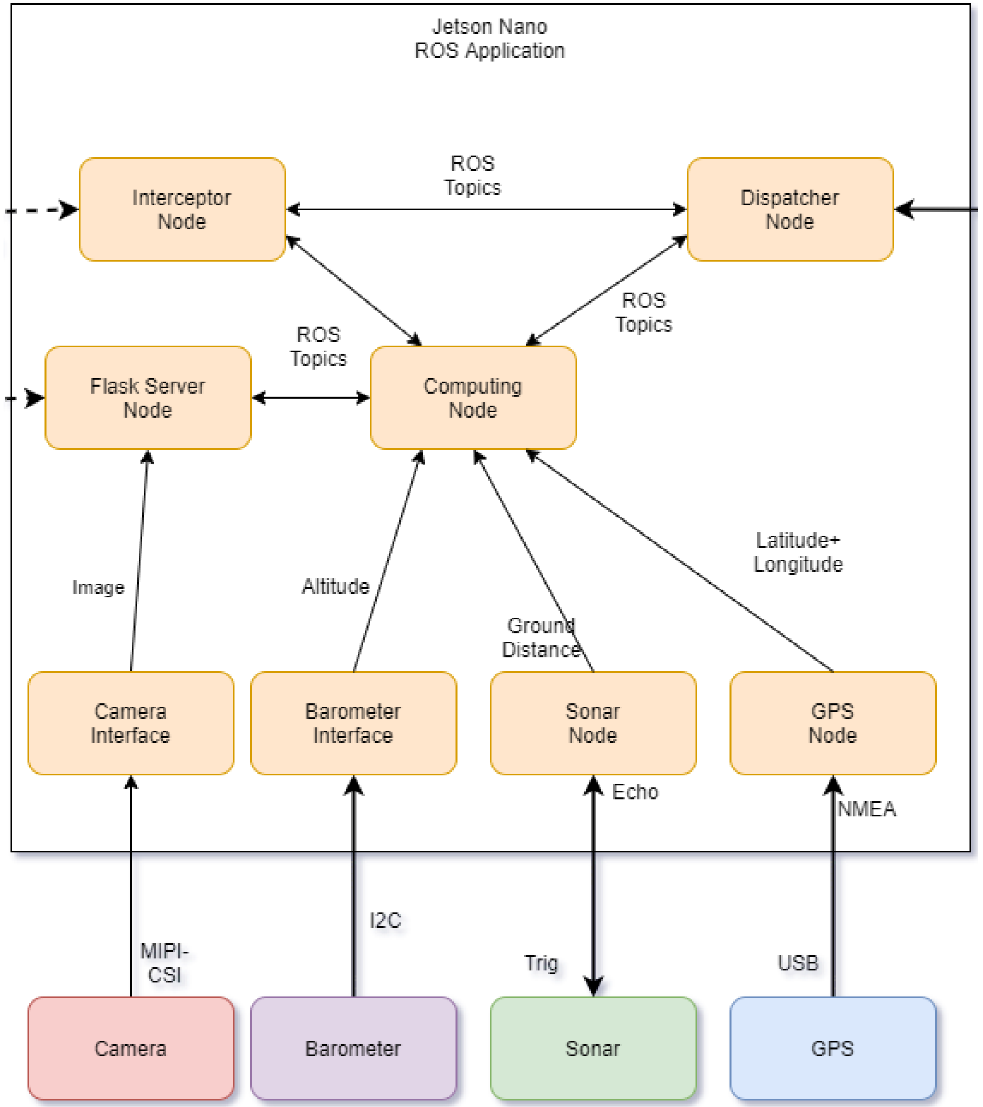


ROS Application

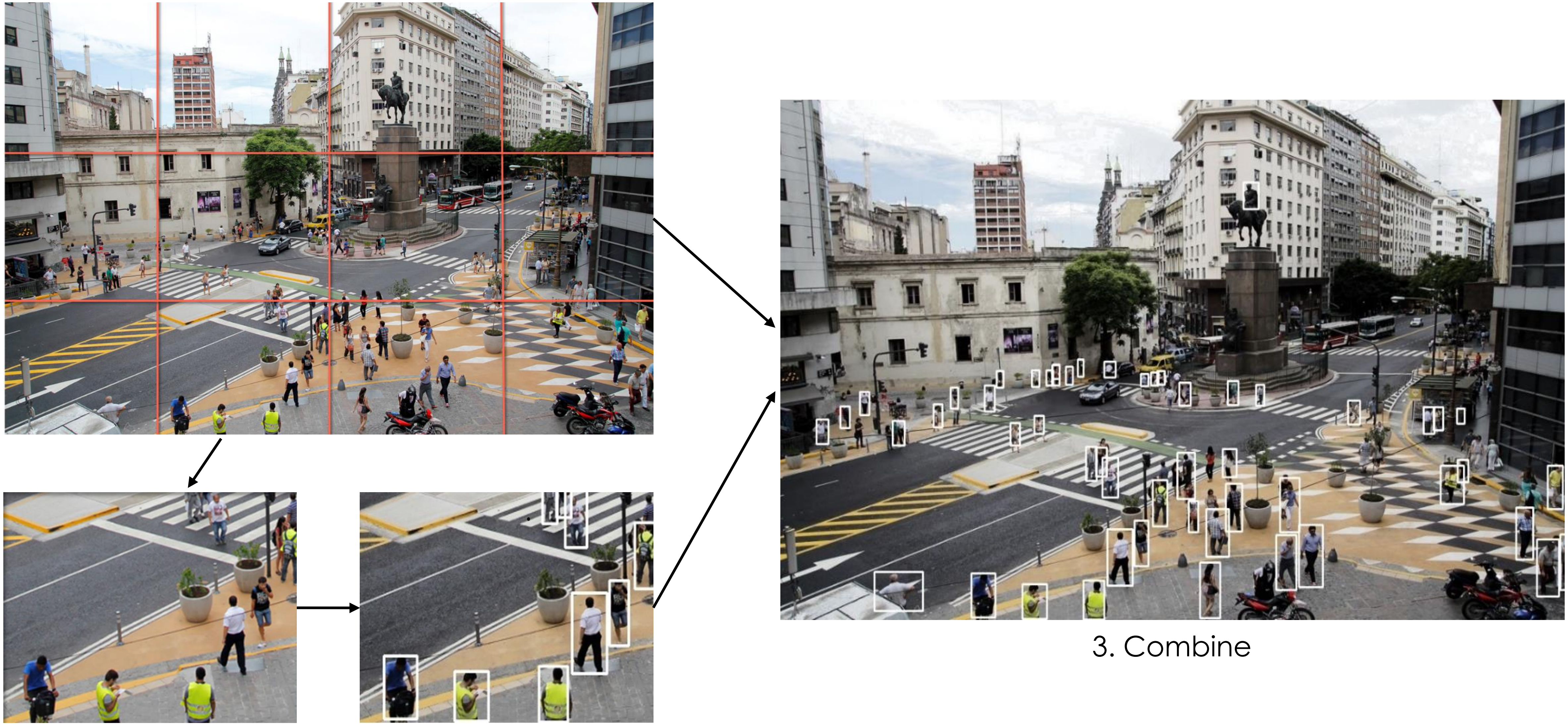
runs on JetsonNano

- MobileNetV2 SSD for person detection
- Overcrowding & social distancing
- Data gathering & logging to .csv
- Flight control via MSP

→ Parallel execution



Person detection

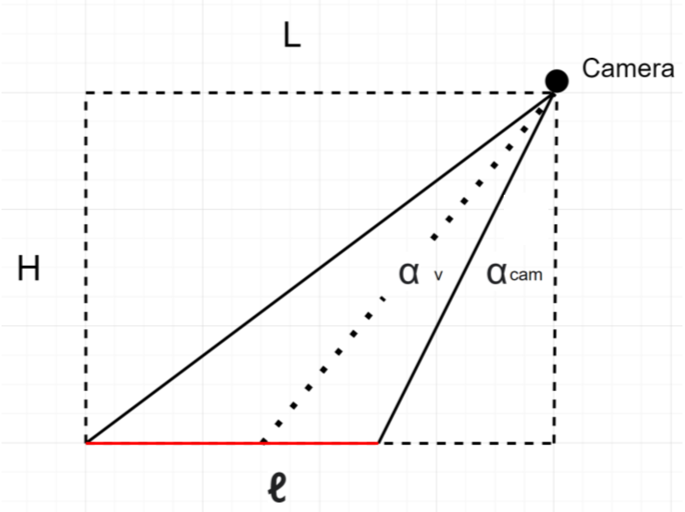


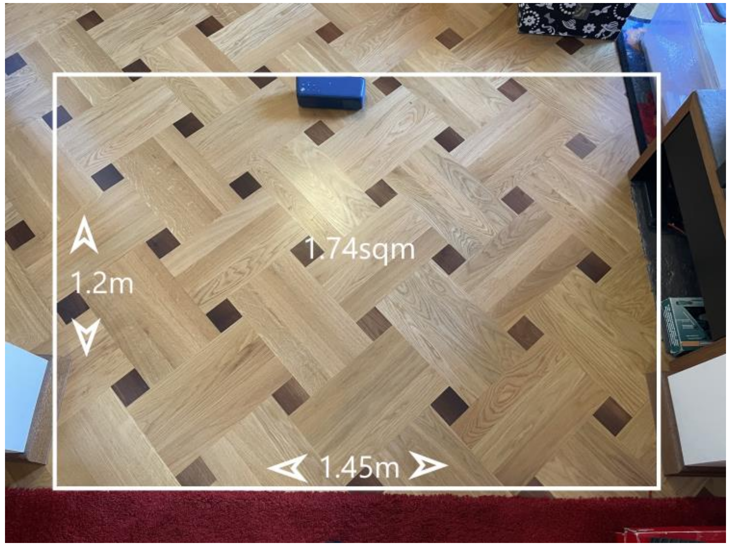
1. Tile

2. Detect

3. Combine

Overcrowding & Social distancing





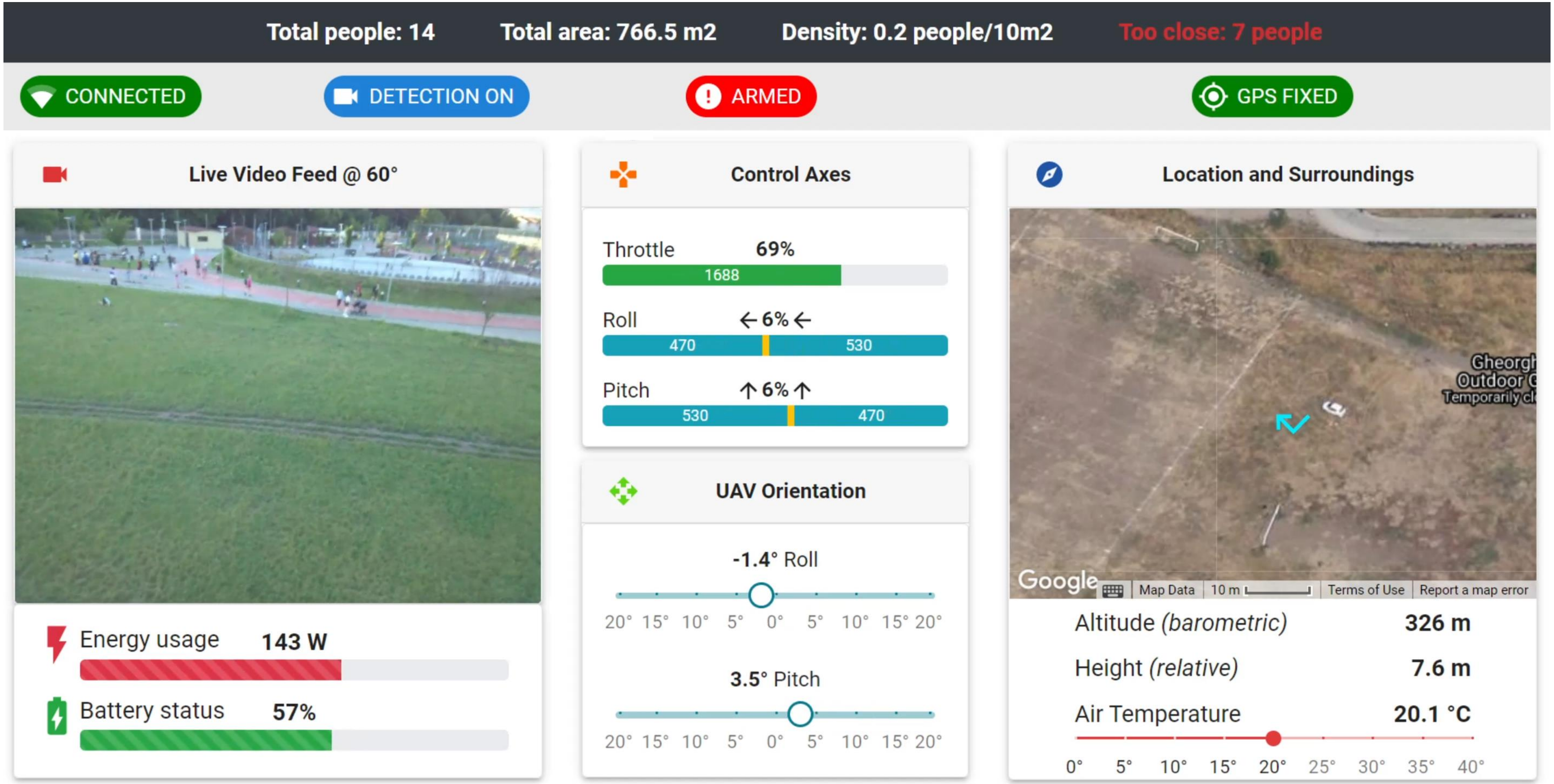
Area estimation test

➡ Estimate distances between people

➡ Estimate average density of crowd

- Density = Area / People
- Obtain area of ground projection from H and α_{cam}
- Pixel size = Area / resolution [cm]
- Distance on ground = Distance on image * pixel size

Client Application *runs on GS Computer*



Total people: 14 Total area: 766.5 m2 Density: 0.2 people/10m2 Too close: 7 people

CONNECTED DETECTION ON ARMED GPS FIXED

Live Video Feed @ 60°

Energy usage 143 W

Battery status 57%

Control Axes

Throttle 69% 1688

Roll ← 6% ← 470 530

Pitch ↑ 6% ↑ 530 470

UAV Orientation

-1.4° Roll

3.5° Pitch

Location and Surroundings

Altitude *(barometric)* 326 m

Height *(relative)* 7.6 m

Air Temperature 20.1 °C

Control application

runs on GS computer

- Sends UDP messages to the ROS application

- Message:
  4 control axes (Roll, Pitch, Throttle, Yaw)
  2 triggers
  6 buttons
  1 Directional pad

- Controls drone & functions

  (flying, camera tilt, detection etc.)

Validation

Board	CPU	TOPS	Interfaces	Speed	Weight	Cost	PricePerf
RPi4	4	0	USB2+3,HDMI, UART, I2C, SPI	483ms	46g	$50	24,150
RPi4 + Coral	4	4	USB2+3, UART, I2C, SPI	25ms	~100g	$110	2750
Jetson Nano	4	0.5	USB3, DP, HDMI, UART, I2C, SPI	25ms	250g	$100	2500
Jetson Xavier	6	21	USB3, DP, HDMI, UART, I2C, SPI	1.25ms	400g	$400	500

Main computer comparison using MobileNetV2SSD inference speed

N	M	Tile Width	Tile Height	Runtime/Tile	Runtime/Image	Correctness
3	2	544	616	50ms	305ms	65%
4	3	408	410	54ms	650ms	67%
5	4	326	308	45ms	905ms	73%
6	4	272	308	43ms	1050ms	76%
8	6	204	205	44ms	2100ms	67%

Image tiling comparison for 1632x1232 input image

Results

Flight deployment at Baza Sportivă Gheorgheni

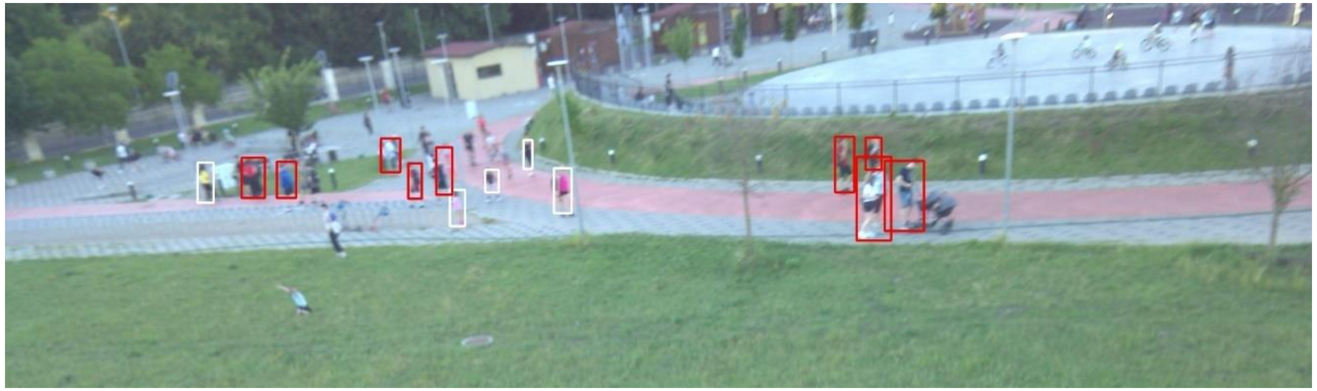
We chose an empty field far from people:

Drone in flight

Test area

Results

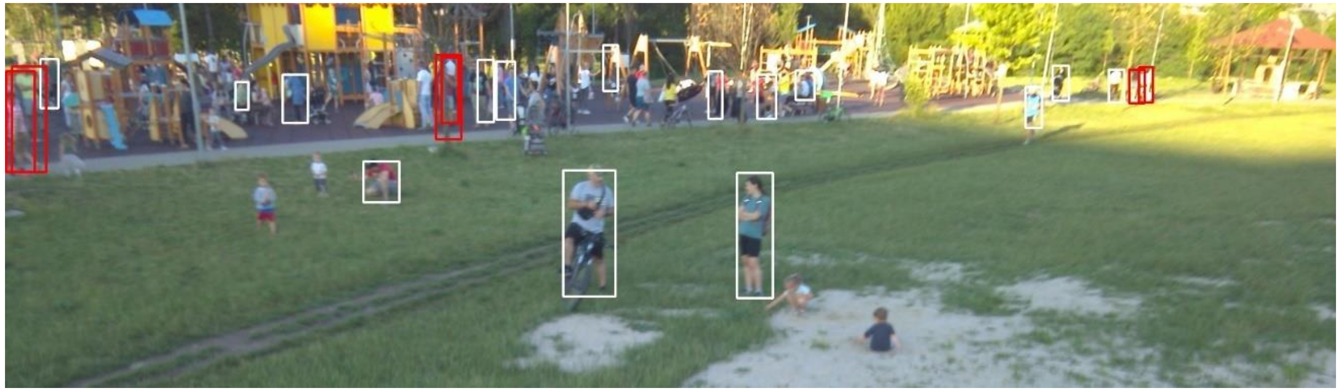
Flight deployment at Baza Sportivă Gheorgheni



7.7m high, **60deg** camera angle

14 of 23 people detected (60.8%)

9 too close, **0.18/10m^2** density in **1340m^2**



3.2m high, **65deg** camera angle

21 of 32 people detected (65.6%)

6 too close, **0.16/10m^2** density in **786m^2**

Conclusion

Complete system for crowd identification and monitoring using UAVs

Real-time overcrowding and social distancing estimations

Low-latency edge processing close to sensors