Project Intelligent Systems: Drones with Artificial Systems

Jatender Singh Jossan, Theodor Bloch, Puneet Singh Roopra
Frankfurt University of Applied Sciences

Project goal

- Develop and build a low-cost FPV drone
- Integrate an AI application (e.g. image recognition)
- Research and possibly integrate an autopilot feature
- Research and possibly integrate a delivery/payload feature

The Drone

Figure 1: Drone with Raspberry Pi and Coral

General Information

- The drone has a 5-inch frame and uses a SpeedyBee flight controller (FC) for stable and responsive flight control.
- A Raspberry Pi is mounted on the drone and equipped with a TPU (Tensor Processing Unit) to handle AI tasks like image recognition.
- The flight controller is connected to a Raspberry Pi via USB, enabling communication between flight systems and onboard processing.
- The remote controller is used to steer the drone and enable the onboard functionalities, powered by the Raspberry Pi.
- A basic autonomous mode has been implemented where the drone flies forward without user input – useful for testing [1], [2].
- The onboard system can analyze images and video in real time, enabling object detection or scene understanding.
- Instead of sending data wirelessly to a ground station, the Raspberry Pi acts as a mini-computer that processes everything directly onboard.

Object detection

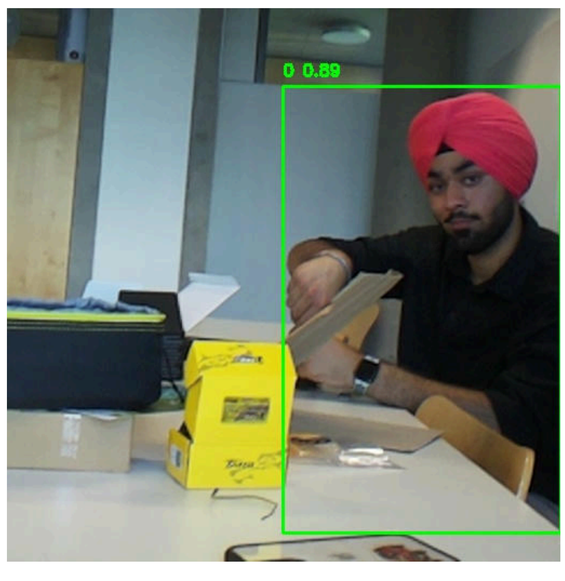

Figure 2: Object detection on an image

Architecture

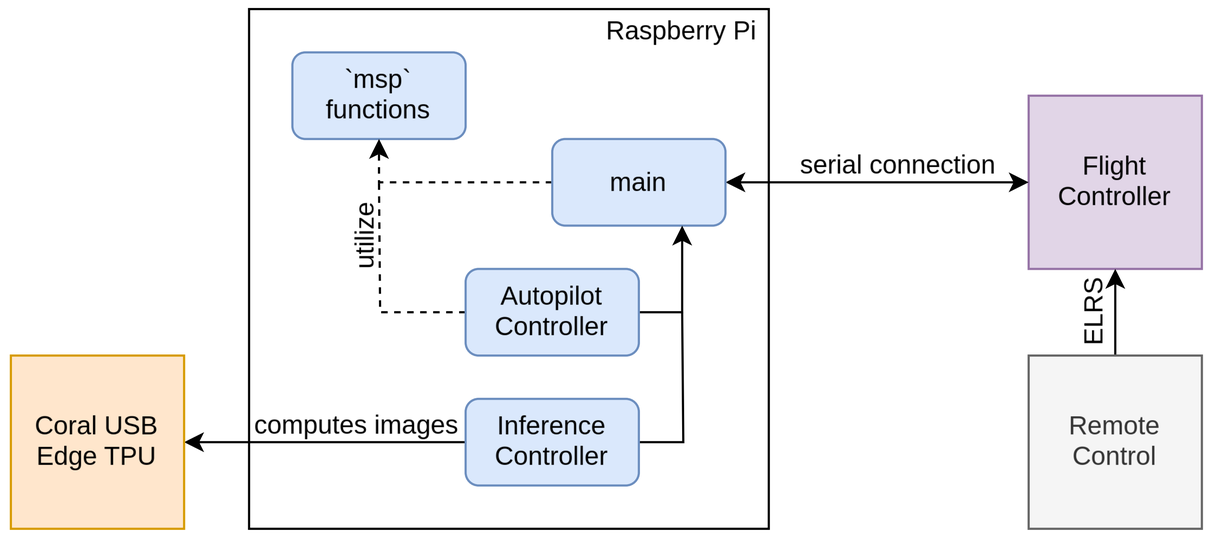

Figure 3: Architecture diagram

Running inference

```python
def process_frame(self, frame):
        common.set_input(self.interp, frame)
        self.interp.invoke()
        objects = detect.get_objects(self.interp, threshold=0.7)
        labels = self.still_labels

    for obj in objects:
        bbox = obj.bbox
        cv2.rectangle(frame, (xmin,ymin), (xmax,ymax), (0,255,0), 2)
        cv2.putText(frame, f'{obj.id} {obj.score}', (xmin,ymin - 10))
        print(f"Detected: {labels.get(obj.id, obj.id)}")

    return frame
```

Listing 1: Processing image/frame [3]

Further work

- Extend the simple Betaflight-based approach with a targeting mechanism based on object tracking (e.g., using OpenCV).
- Build an Ardupilot-based autopilot [4]: Open-source autopilot software compatible with the SpeedyBee F405 AIO, featuring MAVproxy ground station, MAVlink protocol for drone communication, and SDKs that could be integrated with AI functionality.
- Extending the hardware capabilities: Adding a modular, lightweight gripper arm controllable via Raspberry Pi or FC as a potential autonomous extension.
- Exploring different-sized drone builds and analyzing the advantages and disadvantages in terms of speed, agility, and payload capabilities.

Bibliography

[1] D. Sazonov, *Autopilot for FPV Combat Drone on Betaflight (Empty version).* (Feb. 26, 2025). Accessed: Jul. 06, 2025. [Online]. Available: https://github.com/under0tech/autopilot_bee_ept/

[2] R. de Azambuja, *YAMSPy*. (Oct. 06, 2021). Accessed: Jul. 06, 2025. [Online]. Available: https://github.com/thecognifly/YAMSPy

[3] Google LLC, "PyCoral API overview." Accessed: Jul. 06, 2025. [Online]. Available: https://coral.ai/docs/reference/py/

[4] ArduPilot Dev Team, "ArduPilot Documentation." Accessed: Jul. 06, 2025. [Online]. Available: https://ardupilot.org/ardupilot/

Links

Project documentation

GitHub repository

Module: Project Intelligent Systems (Summer Term 2025) in Allgemeine Informatik (M.Sc.)
Proctor: Prof. Dr. Christian Baun