First Class Flights | Autonomous Control of UAVs

Cadets Ralis, Huang, Kelly, and Bartmann

Overview

- Introduction
- Background / Definitions
- Motivation and Problem
- Abstract
- General Materials
- Results
- Demo / Simulation Results
- Progress Made / Conclusion
- Challenges and Limitations
- Future Works

Background / Definitions (1)

- **Unmanned Aircraft System (UAS)** - refers to the entire system in the operation of an unmanned aerial vehicle (**UAV**), which itself is a component of the UAS which also includes things such as modules, sensors, software, etc.
- **Microcontroller** – a small, specialized computer designed to handle specific, low-level tasks and logic.
- **Microcomputer** – similar to a microcontroller, however it can be thought of exactly as its namesake implies: just like a normal computer but in a small footprint; microcomputers are more general purpose, typically running a traditional operating system. (Raspberry Pi is an example of a microcomputer)

Background / Definitions (2)

- **Light Detection and Ranging (LIDAR)** - a remote sensing method for determining ranges by emitting a pulsed laser and evaluating the time it takes for the laser to be reflected off of objects back to a receiver.
- **Transmission Control Protocol (TCP)** - "a communications standard that enables application programs and computing devices to reliably exchange messages over a network."
- **RSA** – an encryption algorithm which utilizes a public and private key to asymmetrically encrypt and secure communications over networks.
- **Machine Learning** - the process of using models, algorithms, and data to help a computer learn autonomously through pattern detection and inference.

Motivation and Questions

- The proliferation of Unmanned Ariel Systems (UAS) has had a profound effect on commercial practices, in individual entertainment worldwide, and military uses. The remote nature of UAS permits a ratio of drones to pilots that is far greater than one. It becomes necessary to move away from a traditional controller scheme and innovate.
- How do you control several drones as one person?
- What sacrifices must be made to allow this functionality?
- How do we overcome these shortcomings?
- What's the best way to make this user friendly?

Abstract

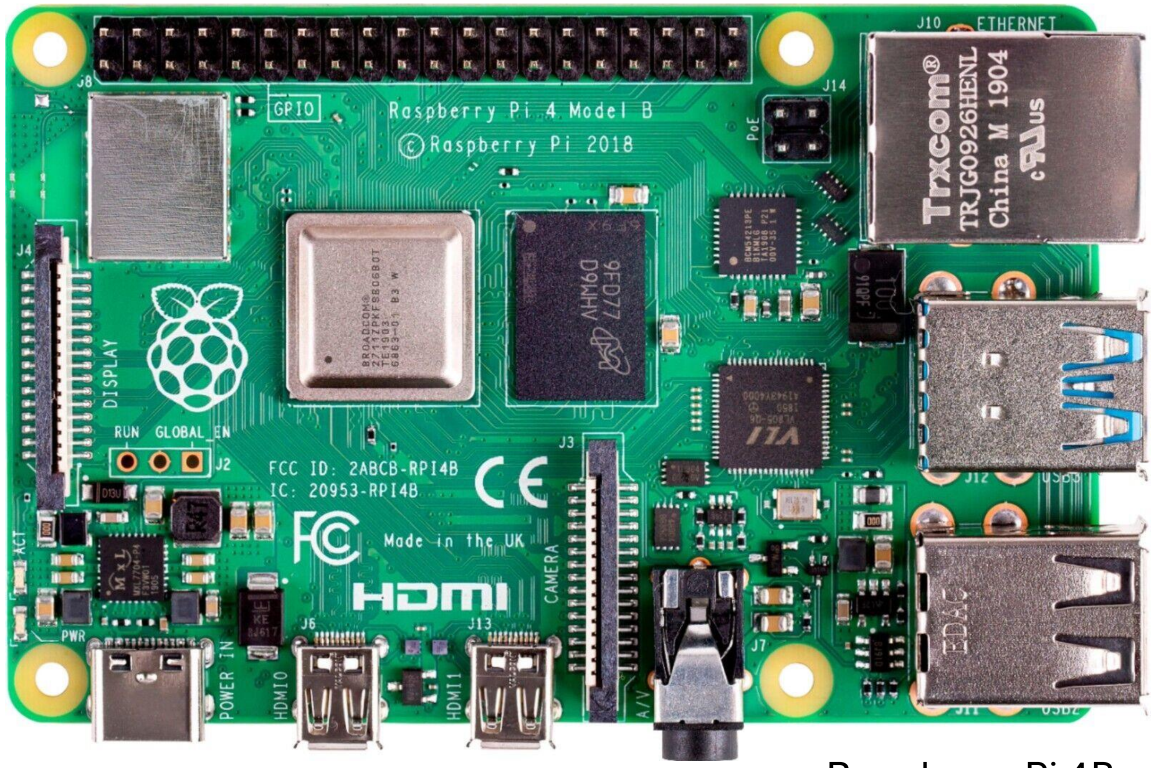
This project explores the development of an autonomous drone system using a standard frame kit and off-the-shelf cameras and sensors. By combining pre-trained machine learning models with onboard processing, the drone can detect and track a person based on simple visual cues—like wearing white or raising fingers on both hands. The system moves beyond traditional manual controls, aiming for more flexible, intelligent behavior. This prototype highlights how low-cost drones can support smart tracking and lays the groundwork for future systems where one operator could oversee multiple UAVs

General Materials

- 1x Raspberry Pi 4 B
- 1x PixHawk Flight Controller
- 1x 1080p ELP USB Camera
- 1x TF-mini plus LIDAR
- 1x F-450 Drone Body
- 4x Brushless Motors
- 4x Motor Electric Speed Controllers
- LiPo Battery + Dis/Charge Controller

Raspberry Pi

- Fully functional “Desktop” Computer
- 4-8 GB of Ram
- 1.5 GHZ Quad Core CPU
- 2 Mini-HDMI Ports
- 5 Total USB Ports
- Ethernet and WiFi (with extra parts)
- GPI/O and other functional pins



Raspberry Pi 4B

1080p Camera

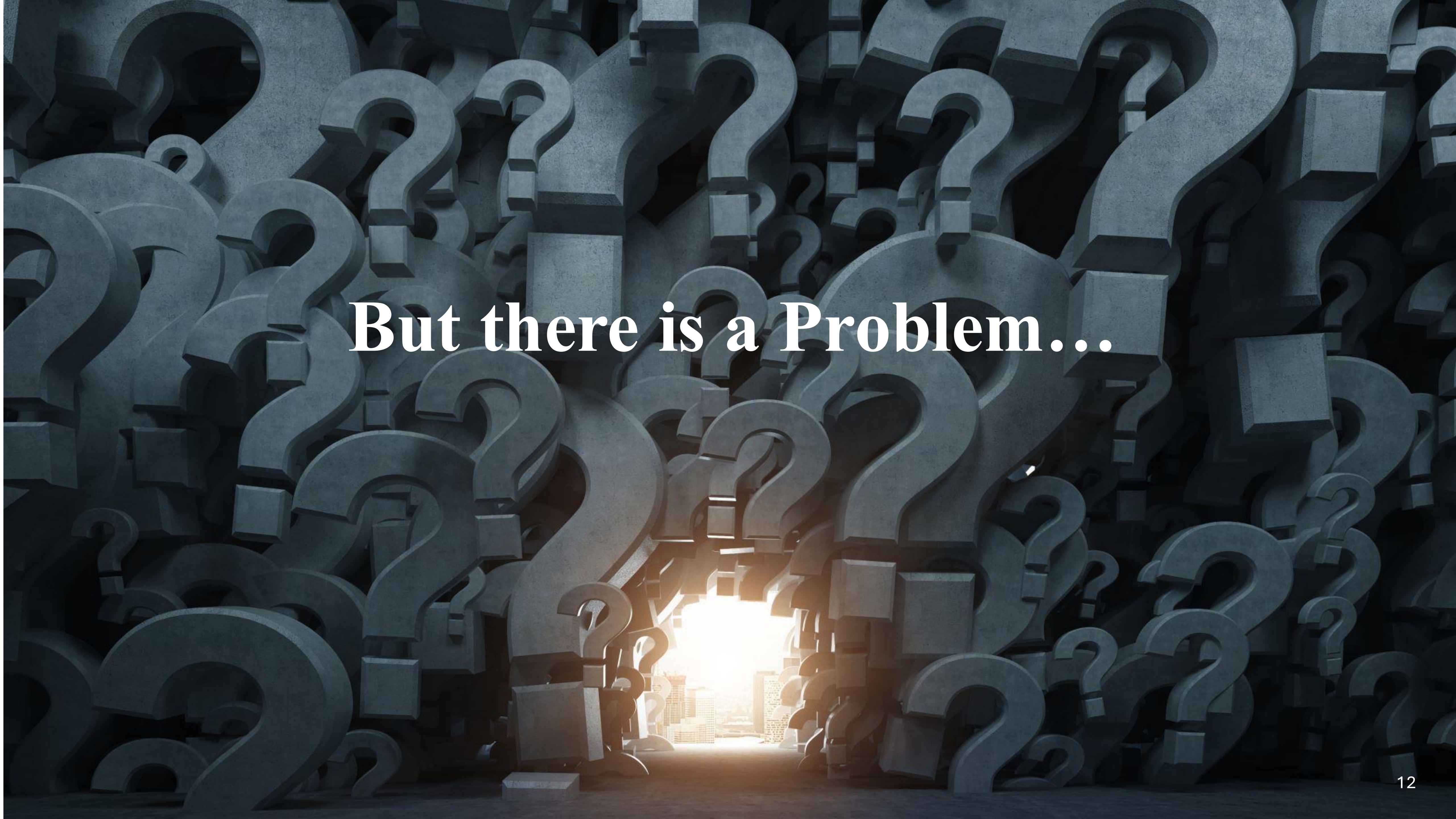
- Made by ELP
- Variable Refresh Rate
  - 30, 60, 100 fps
- 3.6 mm lens (seems adequate)
- Day/night vision
- USB connection

ELP 2MP high refresh rate USB camera

LiDAR

- High sensor refresh rate
- 10-12 meter range
- Functionality
  - ~~Auto takeoff and landing~~
  - Spatial awareness/telemetry
  - Limiting movement through close rangefinding
- Interface via UART

TF LiDAR mini device sensor



But there is a Problem…

Results

- Autonomous Flight via GUIDED mode
  - Along pre-planned routes
  - Allow deviations to the flight plan as directed
  - Follow a target once identified
  - ~~Collison avoidance~~
  - ~~Airspace awareness~~
- Used pre-trained ML model
  - Yolo v5 Nano
  - MediaPipe Hand Landmarker

Demo / Simulation Results

- Tracking a person wearing white
- Detecting 10 fingers
- LIDAR for distance management from ground/target
- Terminal outputs or camera view with overlays

Progress Made / Conclusion

- Majority of drone assembly has been completed.
- Researched and familiarized with coding, algorithms, and calibration for autonomous flight, object recognition and tracking.
- Familiarity with Linux Serial, UART, and I2C on Raspberry Pi
- Hands-on with 3D printing for any necessary additions to the existing kit frame.

Challenges and Limitations

- Hardware Limitations
  - Raspberry Pi has limited processing power for real-time detection
  - USB cameras may have lower resolution and occasional lag
- Detection Accuracy
  - Only-Whites detection can be affected by lighting and background
  - Finger tracking may not work well if hands are not clearly visible
- Performance Trade-offs
  - Adding more components increases the weight and power usage of the drone
  - Flight time may decrease due to onboard processing demands
- Current System Limits
  - The drone tracks visually but does not yet respond with movement
  - Obstacle avoidance and flight control are not yet implemented

Future Works

- Ground Station
  - Commands issued from a ground control station
- Autonomous Flight
  - Fly along long pre-planned routes
  - Allow deviations to the flight path once a target is identified
  - Follow/Keep at range a target once identified
- Tracking
  - Another camera facing the ground for better awareness
  - Fisheye lenses

Thank you for your time!

Any questions?