Search and Rescue Drone

Project Proposal by Gaurav Savant, Ankit Khandelwal, and Bhavik Thati

Problem: Search and Rescue is Hard!

Large Human Resources Required

Search and rescue operations often require a large amount of manpower and heavy equipment in order to probe large areas of land or water

Rescue Teams Are Slow

With a time sensitive operation, it is imperative that medical teams find and apply medical attention as fast as possible.

High Cost of Rescue Operations

Rescues will typically need specialized equipment, highly trained personnel, and operations can often take hours leading to skyrocketing costs.

Use Case Requirements

Rapid Target Detection

1. Object detection within range of 10 feet
2. Mission completion (Search area + detect object) in 10 minute for 6000 square yards (size of a football field)
3. LED flashing when mission accomplished
4. Accurate tracking of a moving object moving at up to 10 mph

Use Case Requirements

Drone Flight Controls

1. Precise and controlled movement
2. All PID controllers synchronized and stable for smooth drone use.
3. Need high processing speeds (100 hertz)

Low Materials Cost

1. Overall equipment value stays within $600, leading to a 90% reduction in cost (SAR drones can cost anywhere between 5,000 and 15,000).
2. Use easily accessible materials (plastic, cheap drone motors, open source software, cheap hardware)

Technical Challenges:

Drone Software

Objection Detection

1. Access to training data
2. Accurate bounding boxes

Flight Controls

1. Stabilizing in unstable environments
2. Eliminating drift from sensors

Path Planning

1. Minimizing overlap while also making sure there are no blind spots in the search
2. Deriving direction and speed from camera data.

Technical Challenges:

Drone Hardware

CNN Acceleration

Receiving accurate data from the AI model and sending accurate results back

Power Distribution

Distribute power to Arduino, KRIA, drone motors

Signal Routing and Image Processing

Making sure all camera data signals arrive at the KRIA and any resultant data is routed out of the KRIA to the Arduino

Mechanical Construction

Ensuring motors can support the weight of all the hardware

Solution Approach:

Vision

1. Computer vision Yolo V8 nano Model
2. Training on Google clusters
3. Export using Onnx model weights format

KRIA

1. HLS
2. Camera Interface
3. UART encoder
4. ARM core/FPGA integration with OpenCL hooks.

Flight Control + Path Planning

1. IMU sensor stabilization using Kalman filters
2. Developing projective geometry algorithms to get target direction

Solution Approach

Drone Motors

1. 920kv Hobby Motors (4x 2212 920kv)
2. PWM Motor Controller

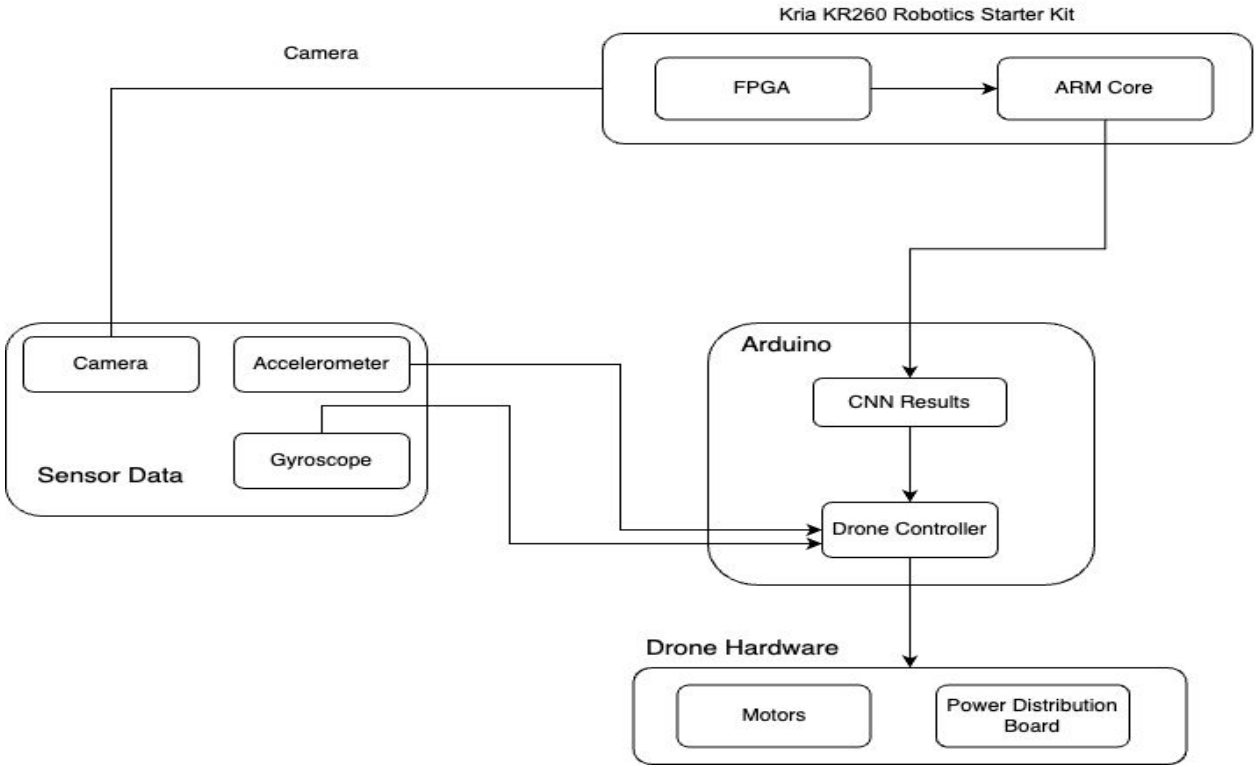
Hardware

1. Arduino Uno for drone controls
2. Gyroscope
3. Altitude Sensor
4. Main MCU
5. Drone Transceiver

Power Management

1. 1000 mAH 3S Lipo Battery
2. Custom power distribution board
3. Does all routing between Kria, Arduino, and associated sensors

System Design Diagram



Testing, Verification, and Metrics

Requirement	Method	Target
Object Detection	Place various objects in front of camera for detection within 20 feet	90% accuracy for object detection
Mission Completion	Start drone search mission in 6000 sq feet area with object in various location	Drone finds object in ≤10 minute for 6000 square yards
System Notification	Place object in front of drone and monitor its led status	Led light blinking 100% of time
Tracking System	Move object slowly (≤ 10 mph) around environment and verify drone can follow	Drone can track/follow object 80% of the time

Testing, Verification, and Metrics

Requirement	Method	Target
Sensor Calibration + Synchronization	Keep IMU on level platform and ensure no drift	See zero drift in roll, pitch, yaw over a minute
Basic Flight Controls	Hang the drone on 3-point harness & hard code set points to move in all directions	Drone can move in all directions
Path Planning Controls	Hardcode basic motion paths and send path control signals to motors	Drone can move in direction as instructed by program

Tasks and Division of Labor

WBS NUMBER	TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION
1	**Drone Mechanical Design**				
1.1	Measure All Onboard Devices	Ankit K	9/15/24	9/20/24	5
1.2	Chassis v1 in Onshape	Ankit K	9/15/24	9/20/24	5
1.3	Print + Test Fit Components on v1	Ankit K + Bhavik T	9/20/24	9/23/24	3
1.4	Chassis v2 in Onshape	Ankit K	9/23/24	9/30/24	7
1.5	Print + Test Fit Component on v2	Ankit K + Bhavik T	9/30/24	10/4/24	4
2	**Sensor Communication + Integration**				
2.1	Wire MPU6050 to Teensy	Ankit K	9/15/24	9/18/24	3
2.2	Wire BMP390 to Teensy	Ankit K	9/15/24	9/18/24	3
2.3	Quantify Drift of MPU6050	Ankit K	9/18/24	9/25/24	7
3	**FPGA Setup**				
3.1	Try Running PyTorch on ARM Chip	Gaurav S	9/15/24	9/23/24	8
3.2	Setup OpenCL Hooks for CNN	Gaurav S	9/23/24	9/27/24	4
3.3	Setup VITIS/HLS	Gaurav S	9/30/24	10/4/24	4
3.4	Develop UART Module	Gaurav S	10/7/24	10/14/24	7
3.5	Develop Image Processing Module	Gaurav S + Bhavik T	10/14/24	10/25/24	11
4	**PCB Design**				
4.1	Finalize Components Needed	Gaurav S	10/7/24	10/7/24	0
4.2	v1 Board Layout Finalized	Gaurav S	10/7/24	10/13/24	6
4.3	v1 Board Sent to PCB House	Gaurav S	10/14/24	10/20/24	6
4.4	v1 Board Testing	Gaurav S	10/21/24	10/25/24	4
4.5	v2 Board Layout Finalized	Gaurav S	10/28/24	11/1/24	3
4.6	v2 Board Sent to PCB House	Gaurav S	11/4/24	11/8/24	4
5	**Software**				
5.1	Train CNN	Bhavik T	9/15/24	9/20/24	5
5.2	Develop PID on Arduino	Bhavik T + Ankit K	9/30/24	10/4/24	4
5.3	Get Motors to Spin via PWM	Bhavik T + Ankit K	9/30/24	10/4/24	4
5.4	Connect to Rx	Bhavik T + Ankit K	10/7/24	10/11/24	4
5.5	Develop Path Planning	Bhavik T + Ankit K	10/14/24	10/25/24	11