

ARTEMIS PRIME



Product summary

WHAT IS THE PRODUCT?

Search and Rescue Vehicle and autonomous
drone that supports individuals during hazards

Real World Functionality

How does the Artemis system function in response to a real-world hazard?



StarLink Satellite



Artemis' Vehicle emulating Wifi to survivors



Drone to autonomously follow Artemis for a birds eye view

WHAT PROBLEM DOES THIS PRODUCT FIX?

What is its magic?

Why now?


Modern Problems - Earthquakes in Turkey and Syria

Turkey earthquake: Deadly new tremor traps people under rubble

🕒 21 February



Turkey–Syria earthquakes 2023

 **World** Africa Americas Asia Australia More

Audio 🔍

The earthquake in Turkey is one of the deadliest this century. Here's why

📄 Press release

3.7 million children in earthquake-affected Syria face catastrophic combination of threats, warns UNICEF Executive Director, following two-day visit

02 March 2023



GETTY IMAGES

Destroyed buildings in Hatay, southern Turkey, after a new earthquake hit the region on Monday

Aftermath of natural disaster

Emergency Situations; Reliable Solutions



Locating survivors in hazards is difficult



Camera with object detection can aid in detecting survivors



Natural disasters often disrupt communication networks and infrastructure



Wifi in car allows people way to contact emergency services or loved ones

WHO IS BUYING ARTEMIS?

Stakeholders?

- Government agencies
- Unicef and Military Contracts

Does the Artemis do the tasks it was meant to do?

Functionality/Feasibility

Tasks:

- Rover that emits wifi
- Autonomous Drone Following
- Object detection
- Mapping application to pinpoint locations

Solutions:

- Set up a wifi access point on the Raspberry Pi.
- Fit rover with raspberry pi.
- Train ML models to detect survivors
- Calculate location of detected survivors on map

Implementation

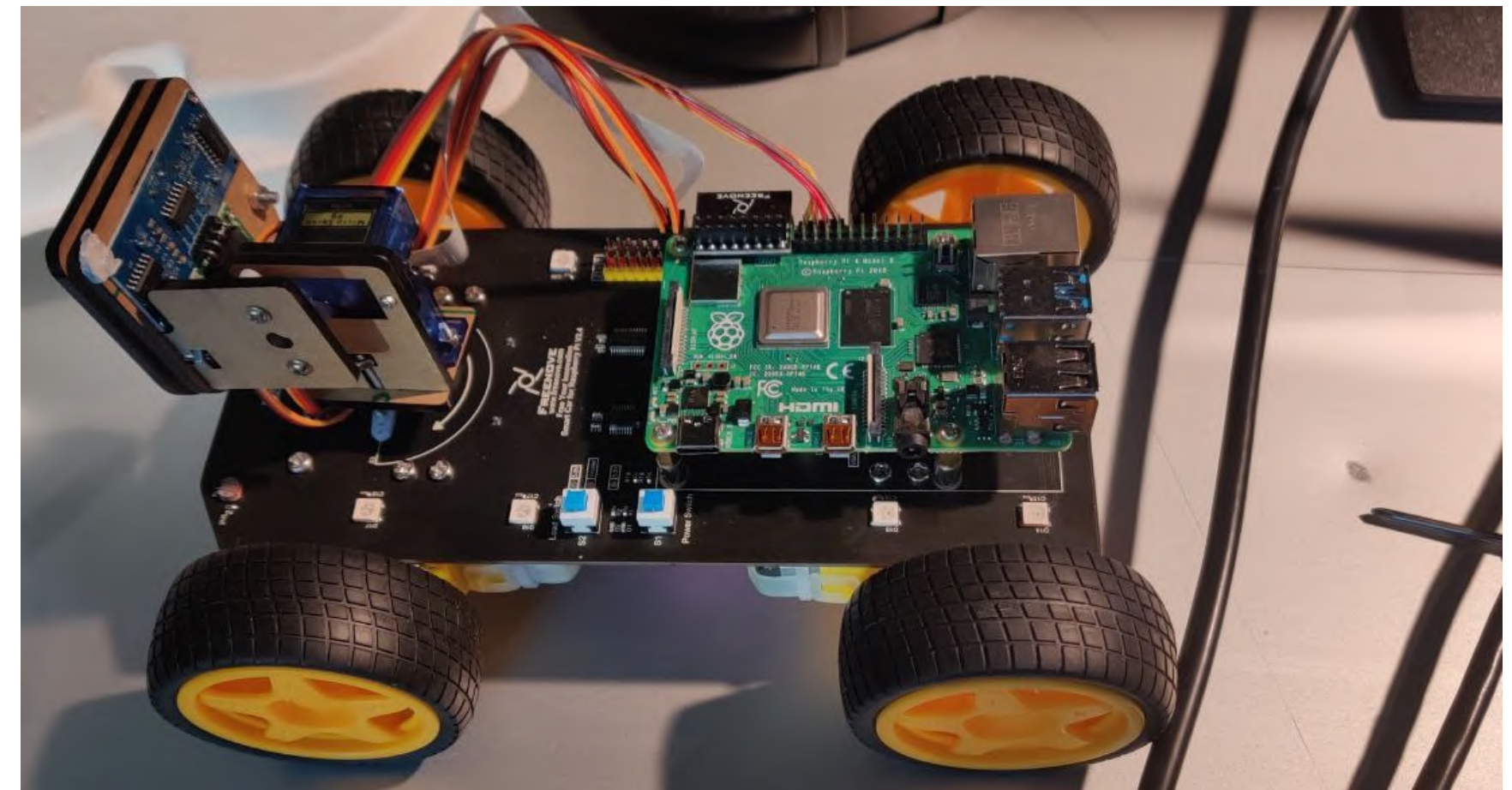
Is it reasonable to implement this project in the real world?

1. Easy method to find survivors that have access to their phones and can connect to Artemis' internet source
2. Useful in hazardous situations like the Turkey Earthquake
3. Offers quick response to prescout minimize lives lost in the aftermath.

Wifi Access Point

HOW TO MAKE VEHICLE WIFI ACCESS POINT

1. Using the raspberry pi's wifi to configure a network UI between the user and raspberry pi
2. Set up a DHCP server and create name
3. Enable the server and reboot the Raspberry Pi.



DRONE AUTONOMOUS FACE DETECTION

Drone Following Face:

Pros:

- Fast and efficient data
- Real-time monitoring
- High degree of accuracy
- Identifies humans



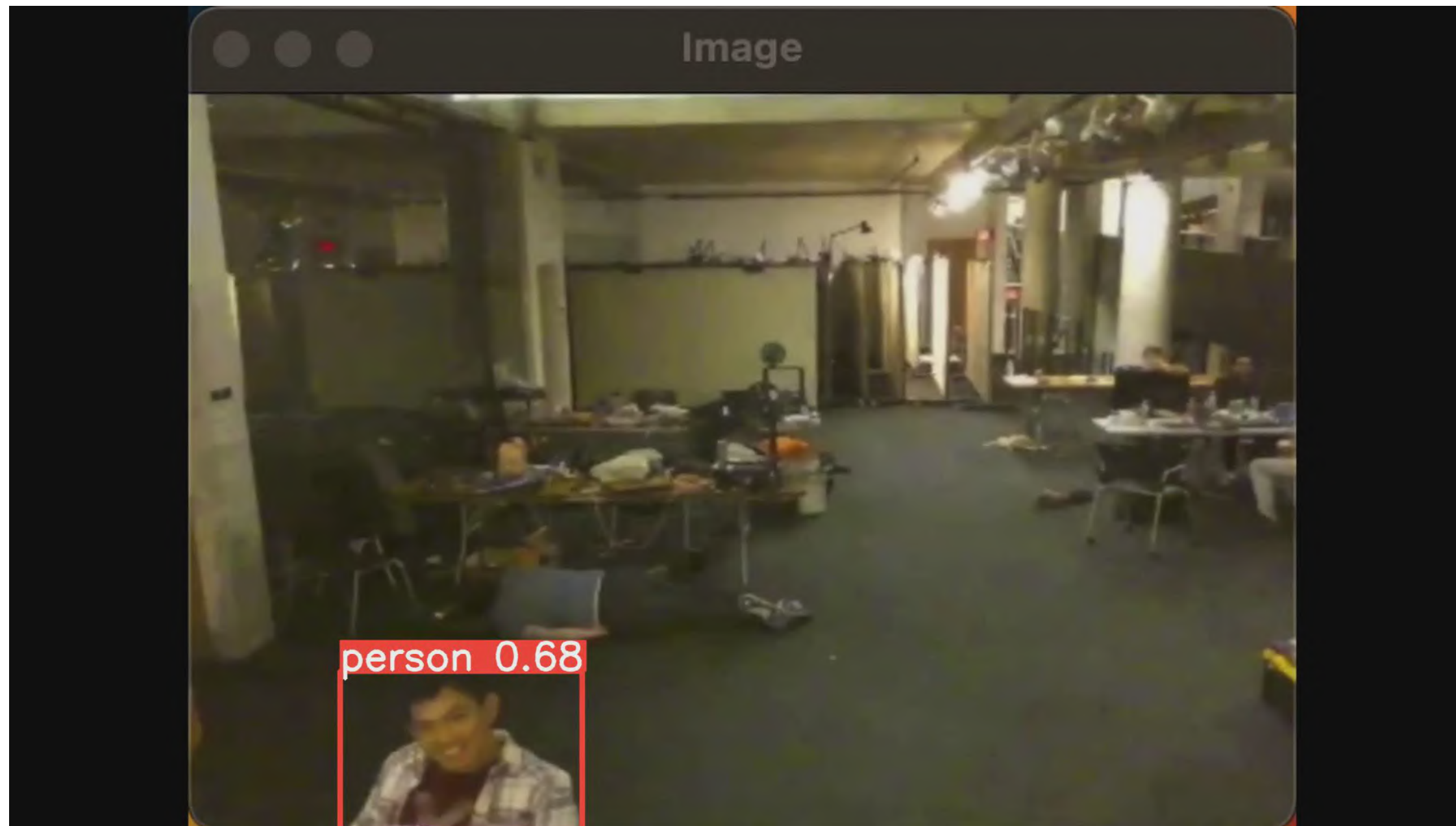
DRONE OBJECT DETECTION

What is the need for drone object detection?

1. Object detection enables drones to automatically identify and track objects of interest
2. Provide an enhanced birds eye view in complex environments
3. Can be trained to locate survivors and hazards

DRONE POINT OF VIEW/OBJECT DETECTION

Live footage from the Drone's point of view:

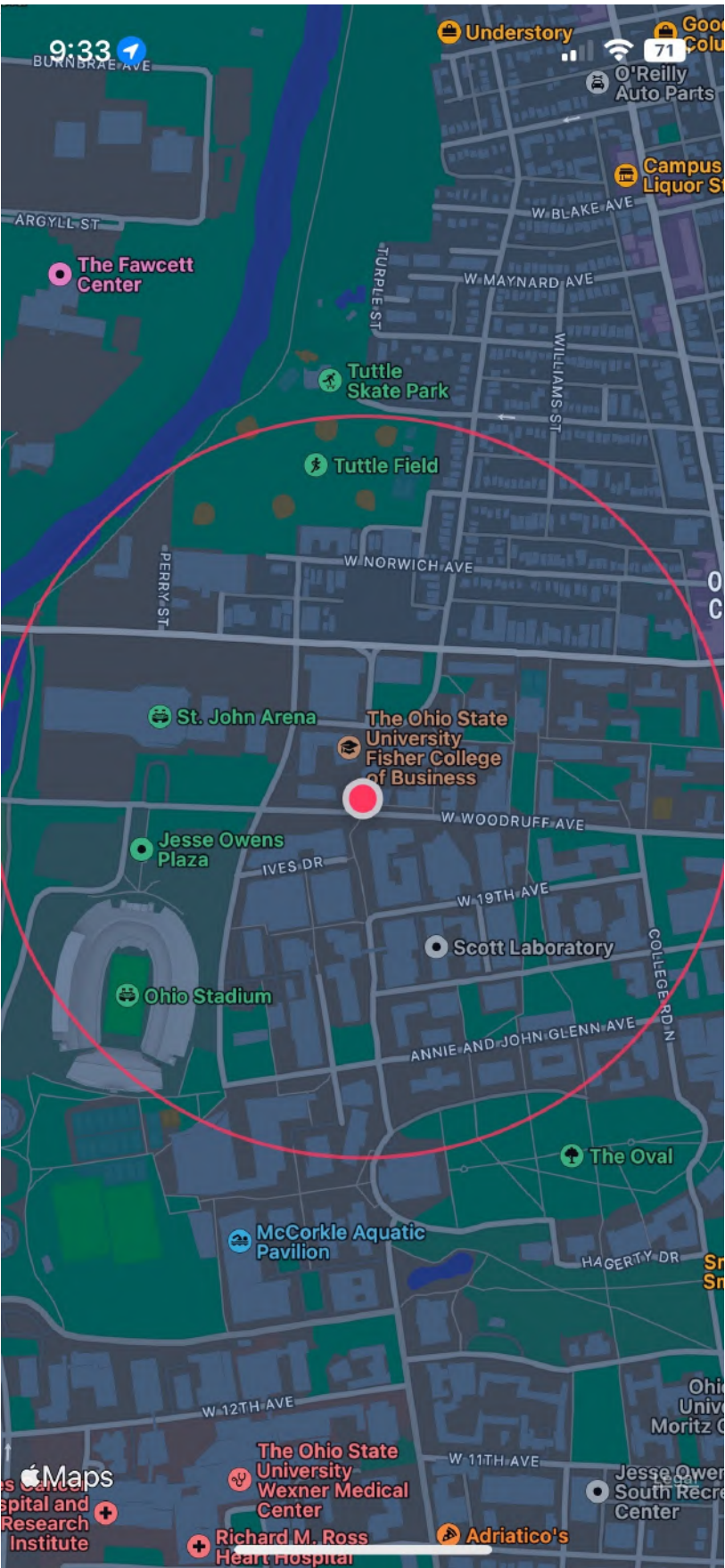


contentView

```
1 //
2 // ContentView.swift
3 // MakeOhioApp
4 //
5 // Created by Pranav Chati on 3/4/23.
6 //
7
8 import MapKit
9
10 import SwiftUI
11
12 struct ContentView: View {
13     @StateObject private var viewModel = ContentViewModel()
14
15
16     var body: some View {
17         Map(coordinateRegion: $viewModel.region,
18             showsUserLocation: true,
19             userTrackingMode: .constant(.follow))
20             .onAppear {
21                 viewModel.checkIfLocationServicesIsEnabled()
22             }
23             .overlay(
24                 Circle()
25                     .stroke(Color.pink, lineWidth: 2)
26                     .opacity(0.8)
27                     .frame(width: 400, height: 400)
28             )
29             .ignoresSafeArea()
30             .accentColor(Color(.systemPink))
31
32     }
33 }
34
35
36 struct ContentView_Previews: PreviewProvider {
37     static var previews: some View {
38         ContentView()
39
40     }
41 }
```

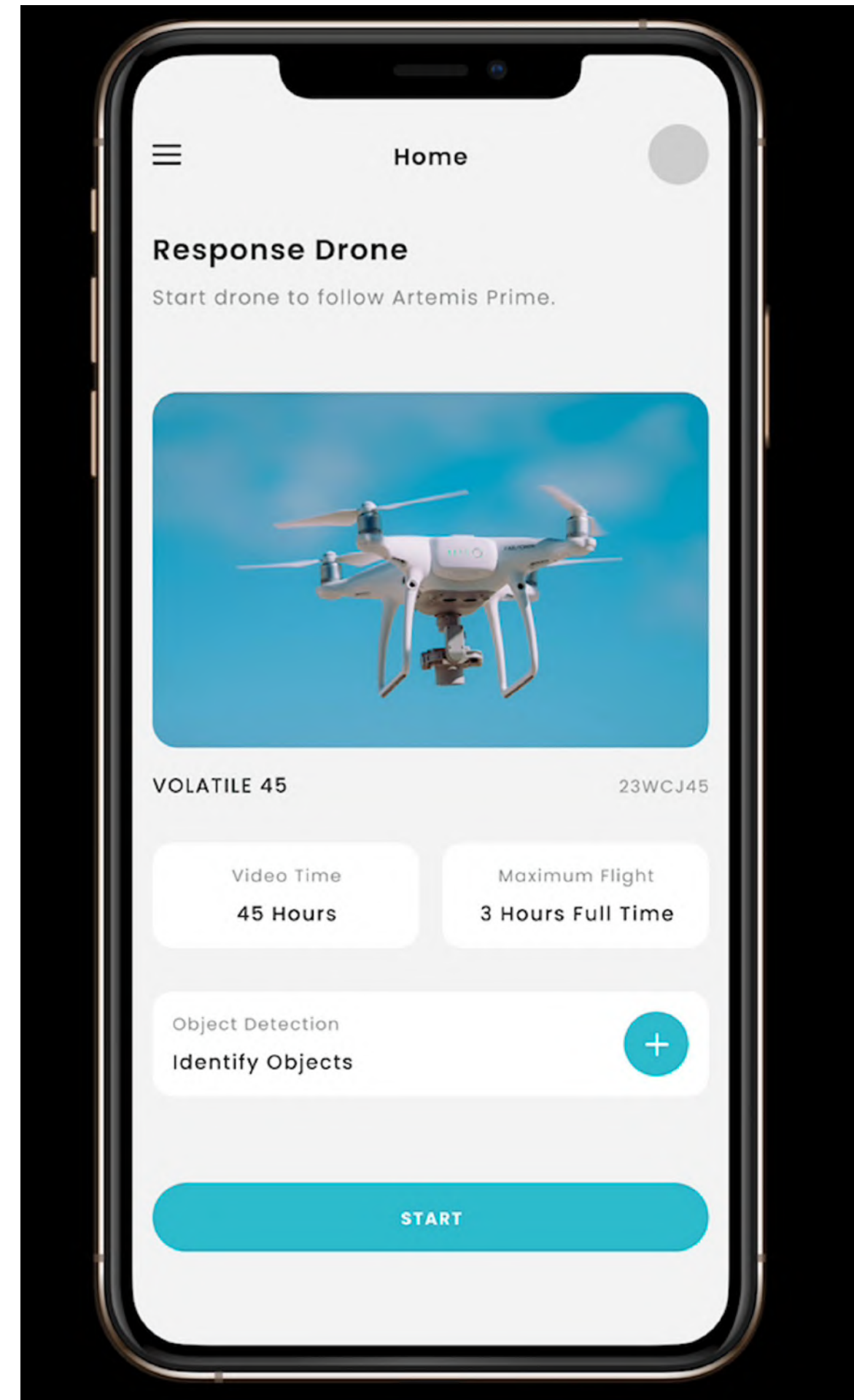
contentViewModel

```
1 //
2 // ContentViewModel.swift
3 // MakeOhioApp
4 //
5 // Created by Pranav Chati on 3/5/23.
6 //
7
8 import MapKit
9
10 final class ContentViewModel: NSObject, ObservableObject, CLLocationManagerDelegate {
11     @Published var userLocation: CLLocationCoordinate2D?
12
13     @Published var region = MKCoordinateRegion(
14         center: CLLocationCoordinate2D(
15             latitude: 39.992981,
16             longitude: -83.001221),
17         span: MKCoordinateSpan(
18             latitudeDelta: 0.01,
19             longitudeDelta: 0.01)
20     )
21
22     var locationManager: CLLocationManager?
23
24     func checkIfLocationServicesIsEnabled() {
25         let status = locationManager?.authorizationStatus ?? .notDetermined
26         if status == .authorizedAlways || status == .authorizedWhenInUse {
27             locationManager = CLLocationManager()
28             locationManager!.delegate = self
29             locationManager?.desiredAccuracy = kCLLocationAccuracyBest
30         } else {
31             print("Location is not authorized")
32         }
33     }
34
35     private func checkLocationAuthorization() {
36         guard let locationManager = locationManager else { return }
37
38         //check for all the cases of the location manager
39         switch locationManager.authorizationStatus {
40             case .notDetermined:
41                 locationManager.requestWhenInUseAuthorization()
42             case .restricted:
43                 print(" Your location is restricted.")
44             case .denied:
45                 print("you have denied your location to be found ")
46             case .authorizedAlways, .authorizedWhenInUse:
47                 region = MKCoordinateRegion(center: locationManager.location!.coordinate,
48                                             span: MKCoordinateSpan(
49                             latitudeDelta: 0.01,
50                             longitudeDelta: 0.01 )
51                 )
52
53                 break
54             @unknown default:
55                 break
56         }
57     }
58
59     func locationManagerDidChangeAuthorization(_ manager: CLLocationManager, didChangeAuthorization status: CLAuthorizationStatus) {
60         checkLocationAuthorization()
61     }
62
63 }
```



Driver User Interface

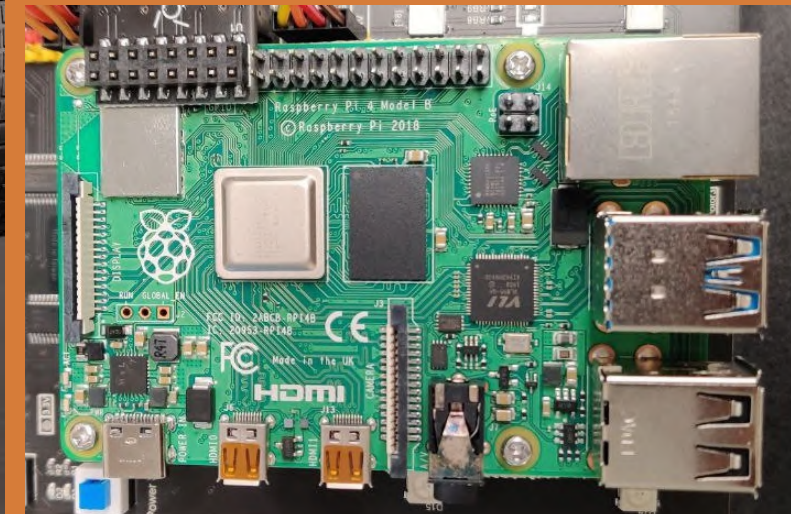
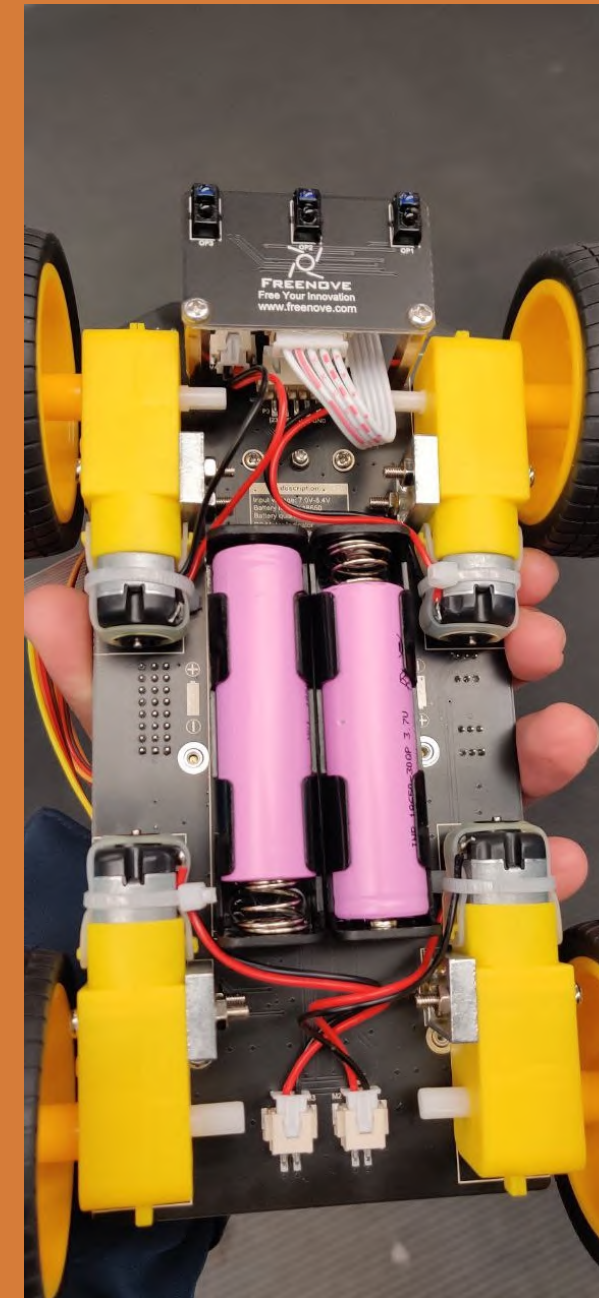
- Created a UI to depict how the app will look like
- Easy-to-use application for driver so they aren't focused on more than one thing when they go to save lives



CHALLENGES WE FACED

Troubleshooting

- Hardware:
 - Lack of components to assemble raspberry pi and rover
 - Poor power supply to power the raspberry pi
 - Bad drone led to lack of bandwidth



FINDING SOLUTIONS

SOLUTIONS

1. Bought batteries from Walmart
2. Condensing rover functionality
3. Getting/Creating a drone that fits the teams requirements better
4. Used object detection on drone feed to deal with low bandwidth

NEXT STEPS

March 2024

Rover Hardware Development

Owing to hardware constraints, the team was unable to fully implement the extensive capabilities of the rover. As a result, the team intends to design a prototype that involves the deployment of a fully autonomous rover with an accompanying drone.

June 2024

Drone Hardware Development + App Development

Due to many constraints of our current drone, it is necessary for us to create a drone that will handle our bandwidth and other security measures and make sure it is regulated and consumer-friendly. The team also plans on developing a fully functional app for drivers to use

January 2025

Launch

By now, the team expects to be launching first official prototypes. We plan on seeking government contracts at this time and implementing this project onto higher class drones and a real vehicle

ARTEMIS PRIME + SAR

THANK YOU