CSE1901 - Technical Answers to Real World Problems (TARP)

Project Report

Autonomous Drone for Military Surveillance

By

19BAI1036 Vaibhav Gadag

19BAI1084 Simrat Singh

B. Tech Computer Science and Engineering

Submitted to

Prof. Geetha S

School of Computer Science and Engineering

April 2022

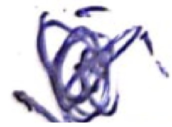
DECLARATION

We hereby declare that the report titled "Autonomous Drone for Military Surveillance" submitted by us to VIT Chennai is a record of bonafide work undertaken by us under the supervision of Prof. Geetha S, School of Computer Science and Engineering, Vellore Institute of Technology, Chennai.

Signature of the Candidate

Simrat Singh

19BAI1084



Vaibhav Gadag

19BAI1036

CERTIFICATE

Certified that this project report entitled "**Autonomous Drone for military surveillance"** is a bonafide work of **Vaibhav Gadag (19BAI1036), Simrat Singh(19BAI1084)** and they carried out the project work under my supervision and guidance for CSE1901 - Technical Answers to Real-World Problems (TARP).

Dr. Geetha S

SCOPE, VIT Chennai]

ACKNOWLEDGEMENT

First and foremost, praises and thanks to God, the Almighty, for His showers of blessings throughout the completion of my project work.

I would like to express my deep and sincere gratitude to my project supervisor, Dr Nithyanandam P, Head of the Department (HoD), B.Tech Computer Science and Engineering. SCSE, VIT Chennai; Dr Ganesan R, Dean of the School of Computer Science Engineering, VIT Chennai; Dr Geetha S, Associate Dean of the School of Computer Science Engineering, VIT Chennai for giving me the opportunity to do this internship and providing invaluable guidance throughout this period. Their dynamism, vision, sincerity and motivation have deeply inspired me. They have taught me the methodology to carry out the projects and to present the research works as clearly as possible. It was a great privilege and honour to work and study under his guidance. I am extremely grateful for what they offered me.

I am also extremely grateful to my parents and friends for their love, prayers, care and sacrifices for educating and preparing me for my future and for their genuine support throughout this research work. I also thank all the staff at Vellore Institute of Technology, Chennai for their kindness and cooperation.

Finally, my thanks go to all the people who have supported me to complete the project directly or indirectly.

ABSTRACT

Our project is to use drones to detect the number of people in the enemy base and give us the information on count so the army can prepare their attack accordingly. We aim to employ drones to monitor no man's land and enforce the protocol, to help maintain the peace. We are building an autonomous drone which runs an object detection algorithm on the video input which it gets through the camera module (pi cam or a mobile phone). The object detection algorithm classifies people and maintains a counter of the number of people detected and assigns every individual person an id number so the person isn't counted twice. We will live stream the drone's video footage to the client-side app. We program the drone to follow a fixed route based on the locale by setting coordinates and checkpoints. Note that these can be altered by the end-user as per requirement.

Link to the Github repository:-
https://github.com/SimratsSingh/TARP-project.git

TABLE OF CONTENTS

CONTENTS

INTRODUCTION

SYNOPSIS

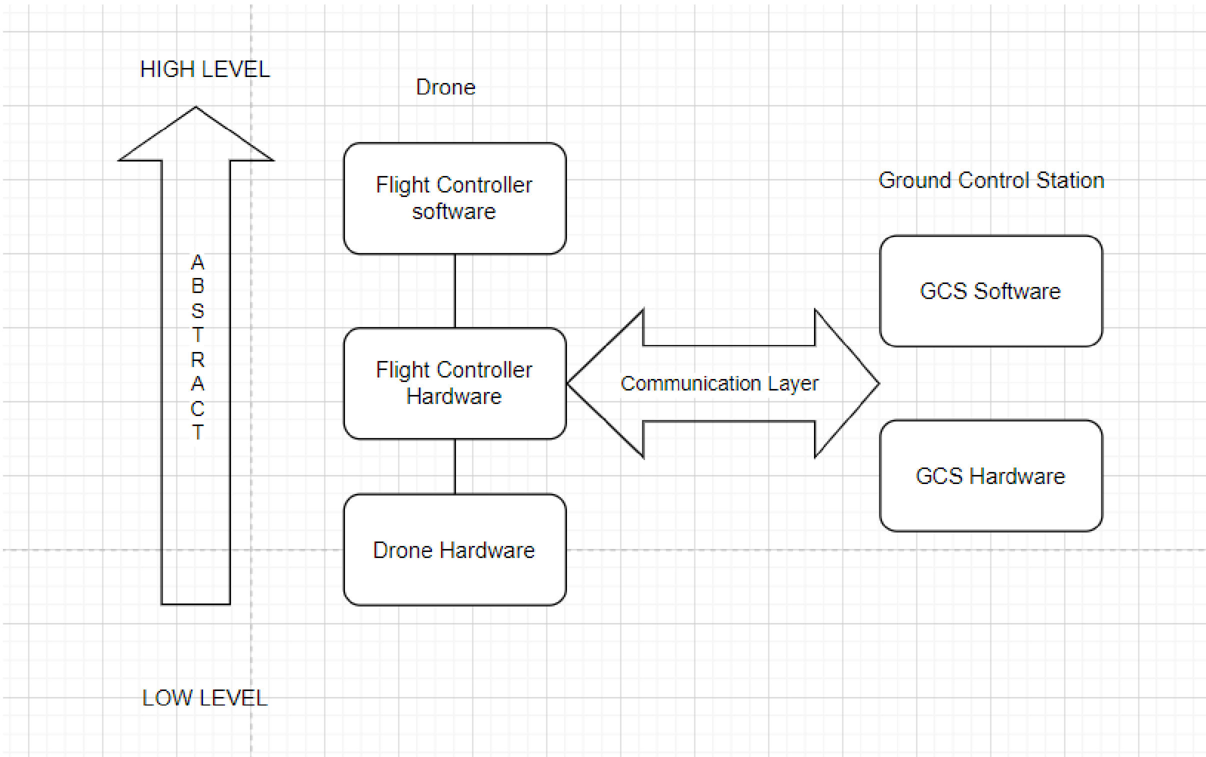
Nowadays national security is the biggest threat to every nation and in places like the border or no mans land it is necessary to implement a surveillance system which can be dynamic and be monitored from a remote location. Hence we came up with the idea of an autonomous surveillance drone.

EXISTING SYSTEM

The existing system for surveillance still include static cctv cameras which can be hacked by entering nto the communication layer, or there are spy cameras and microphones disguised as rocks. The existing systems available are very limited in terms of moving camera angle as well as speed and even accuracy. Our solution helps us to monitor the field from different angles while sitting in a remote location. It helps to give us the pov of all the possible sides and views.

PROPOSED SYSTEM

The system proposed by us include the video feed being transferred in real-time from IoT devices that act as a middleman (in our case raspberry pi or a mobile phone) between the local pc and the camera. We are sending the feed to the local pc as we require high computational power for the processing, whereas we can use a microprocessor like an Nvidea jetson but due to budget constraints we are using the local pc itself. The video feed sent to the local pc is processed by the pre-trained models and then detected if there is a human detected or not. The architecture used for training the model is FasterRCNN. We implemented this on our local pc as we sent the video feed via IP communication.



LITERATURE REVIEW

Drone surveillance can be viewed as either a justified, impartial activity that benefits everyone or as an oppressive tactic that benefits some at the expense of others. This study examines the ethical implications and issues surrounding the deployment of unmanned aerial vehicles, specifically whether citizen monitoring is acceptable. Classic philosophical and behavioural approaches to ethics are employed to answer this topic. The investigation begins with the significance of the problem and then moves on to its evolution and current state. The essay explores reasons for and against domestic monitoring after introducing the technique of analysis. Drones' particular usefulness allows them to do a lot for the public good while also posing moral risks.[1]

The conflict between security and privacy concerns is not new, but it has been exacerbated in recent decades by the rapid advancement of surveillance technologies. Previously restricted to unaided eye inspections, police personnel may soon acquire, or in some circumstances already have, the capacity to see through walls or monitor a person's activities from the air. [3] The question then becomes: What is the correct balance between the government's obligation to keep people safe and citizens' privacy needs? According to certain polls, while the public supports the use of drones in specific situations, they are less enthused about utilising them in normal law enforcement activities. [4][2]

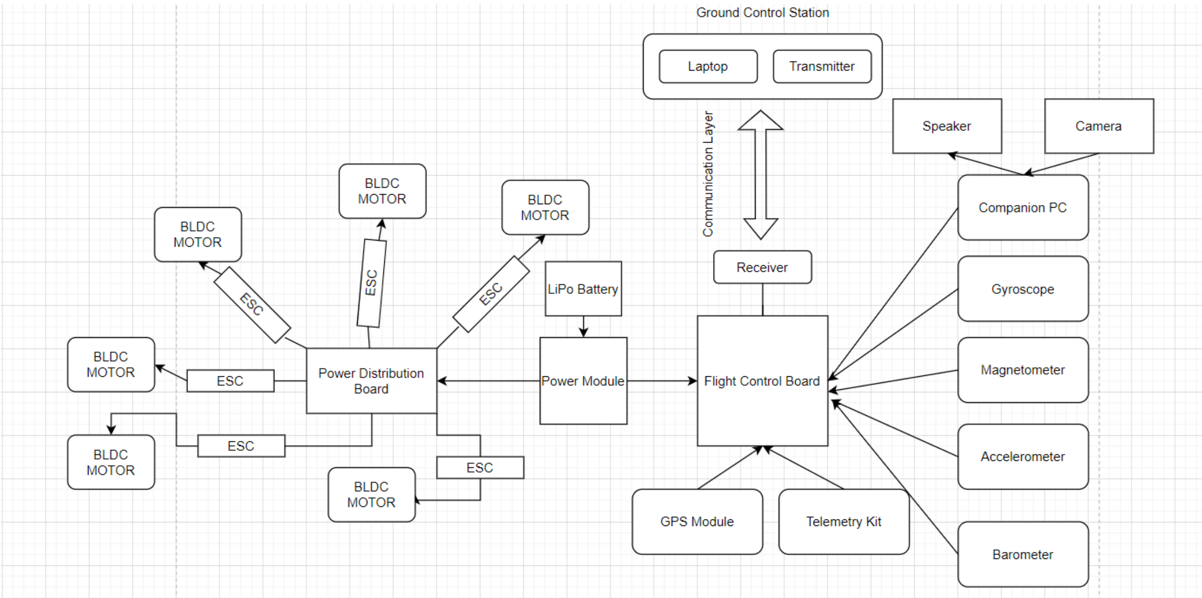
A new class of intelligent sensing systems has been enabled by low-cost mini-drones with enhanced sensing and mobility. To realise the full potential of such drones, newly upgraded formulations of both common and emergent sensing scenarios are required. Specifically, numerous fundamental issues in visual sensing remain unsolved, including [5] fitting large objects into camera frames; [6] locating cameras at effective views matching target positions; and [7] accounting for occlusion by components in the environment, such as other targets.[8]

Aerial surveillance may cover far-flung isolated places, geographically scattered locations, and offshore locations. Drones will be used extensively by oil and gas corporations and utility suppliers, who rely on significant monitoring, measurements, mapping and surveying, and maintenance activities for distant sites. (2) Search and Rescue: Drones can be used on an as-needed basis. Drones can overcome the problem of accessing distant places in an emergency with damaged or unreliable infrastructure, providing speedy transit with considerable ease and flexibility. (3) Risky Missions: Drones are ideal for unmanned missions in risky or hazardous conditions, such as taking measurements at high altitudes or in bio/chemically toxic environments. Drones may be controlled remotely by humans to carry out risky missions.[9]

METHODOLOGY AND ACTION PLAN

Workflow

The workflow for our solution includes the communication between the drone and the local ground station as the most crucial part. The ground station communicates with the drone using the telemetry with the transmitter connected to the local host giving commands using the mission planner software and the receiver connected to the drone. The receiver sends the data to the flight controller which acts as the brain of the whole drone. It controls the motors accordingly to the command given by the mission planner. The video on the side is sent to the local pc via an IP communication protocol.



We needed the component selection for the drone such that we were able to complete the maximum area possible in the minimum amount of time and that the severe temperature conditions do not affect the drone and create problems. Hence choose the following components so that not only the drone gets a 2:1 thrust to weight ratio but also a sturdy body.

Components Table

Name	Model	Weight(grams)
GPS module	Ublox NEO M8N	33
Pixhawk power module	APM/Pixhawk Power Module V6.0	24
ESC	Emax BLHeli Series 30A ESC	28*6
LiPo battery	ORANGE 4200/3S	319
Propellers	Orange HD Propellers 1047(10X4.7)	22*3=66
Drone frame	F550 hexacopter frame	620
Pixhawk	Pixhawk 2.4.8	15
Telemetry	3DR Radio Telemetry	33
Pi Camera	Pi Cam V2	3
BLDC Motor	EMAX xa2212 980kV	55*6=330
Raspberry pi	Model 3b+	50
Total weight		1661 grams

Time of flight calculation :

LiPo Battery used -> 3S 4200mAh 40C

Total weight of the drone = 1661 gm

For achieving 2:1 Thrust to Weight ratio, Thrust needed = 3322gm

Hence, required thrust required per motor = 553gm

According to the datasheet, max possible thrust per motor = 880gm

As per datasheet, 9.6A of the current draw is needed for generating 560gm thrust.

Therefore, average total current draw = 57.6A

Time of Flight (in minutes) = (Battery capacity * 60) / (Current draw)

Time of Flight = (4200 * 0.001 * 60) / 57.6 = 4.375 minutes.

XA 2212 Brushless motor test record							
Motor type	The voltage (V)	Prop. Size	Current (A)	Thrust (G)	Power (W)	Efficiency (G/W)	RPM
XA 2212 820KV	12	APC 11*4.7	12	830	144	5.8	5720
	8	APC 11*4.7	7.3	500	58.4	8.6	4650
XA 2212 980KV	12	APC 10*4.7	15.1	880	181.2	4.9	6960
	8	APC 10*4.7	9.5	550	76	7.2	5470
	12	APC 9*6	12.3	730	147.6	4.9	8220
	8	APC 9*6	7.1	400	56.8	7.0	6090
XA 2212 1400KV	12	APC 8*4	16.4	930	196.8	4.7	12020
	8	APC 8*4	9.1	500	72.8	6.9	8900
	12	APC 8*6	20.6	940	247.2	3.8	10750
	8	APC 8*6	11.9	520	95.2	5.5	8250

Distance covered

The drone moves at a speed of 7m/s

On the basis of that speed drone will be able to cover a path of **1837.5 meters**

The video stream from the Pi-Cam is uploaded to the ground control station system, which runs the server, across a network. The frames are received by the server, which then undergoes additional processing. Message passing, which is also the core notion of distributed systems, is used to do this. Image processing is carried out with the help of an open-source network based on the FasterRCNN architecture.

- Using the FasterRCNN architecture, detect the persons present in the Region of Interest (ROI) and create a bounding box around each person.
- Estimate the distance between each person identified in the frame by using the person's location in the frame as the bottom centre point of the bounding box.
- By applying the transformation to the bottom centre point of each bounding box, the person's location in the bird's eye perspective is approximated.
- Calculate the bird's eye view distance and scale the distances in the horizontal and vertical directions using the calibrated scaling factor.
- Draw bounding boxes around the persons in the frame that have been detected.

Code for Human Detection

Loading the model and defining the classes

```python
import cv2
import datetime
import imutils
import numpy as np
from centroidtracker import CentroidTracker

protopath = "MobileNetSSD_deploy.prototxt"
modelpath = "MobileNetSSD_deploy.caffemodel"
detector = cv2.dnn.readNetFromCaffe(prototxt=protopath, caffeModel=modelpath)
#detector.setPreferableBackend(DNN_BACKEND_INFERENCE_ENGINE)
detector.setPreferableTarget(cv2.dnn.DNN_TARGET_CPU)


CLASSES = ["background", "aeroplane", "bicycle", "bird", "boat",
           "bottle", "bus", "car", "cat", "chair", "cow", "diningtable",
           "dog", "horse", "motorbike", "person", "pottedplant", "sheep",
           "sofa", "train", "tvmonitor"]

tracker = CentroidTracker(maxDisappeared=80, maxDistance=90)


def non_max_suppression_fast(boxes, overlapThresh):
    try:
        if len(boxes) == 0:
            return []

        if boxes.dtype.kind == "i":
            boxes = boxes.astype("float")

        pick = []

        x1 = boxes[:, 0]
        y1 = boxes[:, 1]
        x2 = boxes[:, 2]
        y2 = boxes[:, 3]
```

Implementing the bounding boxes

```python
        pick = []

        x1 = boxes[:, 0]
        y1 = boxes[:, 1]
        x2 = boxes[:, 2]
        y2 = boxes[:, 3]

        area = (x2 - x1 + 1) * (y2 - y1 + 1)
        idxs = np.argsort(y2)

        while len(idxs) > 0:
            last = len(idxs) - 1
            i = idxs[last]
            pick.append(i)

            xx1 = np.maximum(x1[i], x1[idxs[:last]])
            yy1 = np.maximum(y1[i], y1[idxs[:last]])
            xx2 = np.minimum(x2[i], x2[idxs[:last]])
            yy2 = np.minimum(y2[i], y2[idxs[:last]])

            w = np.maximum(0, xx2 - xx1 + 1)
            h = np.maximum(0, yy2 - yy1 + 1)

            overlap = (w * h) / area[idxs[:last]]

            idxs = np.delete(idxs, np.concatenate(([last],
                                                   np.where(overlap > overlapThresh)[0])))

        return boxes[pick].astype("int")
    except Exception as e:
        print("Exception occurred in non_max_suppression : {}".format(e))
```

Implementing the bounding box only if confidence greater than 50%

```
while True:
    ret, frame = cap.read(1)
    frame = imutils.resize(frame, width=600)
    total_frames = total_frames + 1

    (H, W) = frame.shape[:2]

    blob = cv2.dnn.blobFromImage(frame, 0.007843, (W, H), 127.5)

    detector.setInput(blob)
    person_detections = detector.forward()
    rects = []
    for i in np.arange(0, person_detections.shape[2]):
        confidence = person_detections[0, 0, i, 2]
        if confidence > 0.5:
            idx = int(person_detections[0, 0, i, 1])

            if CLASSES[idx] != "person":
                continue

            person_box = person_detections[0, 0, i, 3:7] * np.array([W, H, W, H])
            (startX, startY, endX, endY) = person_box.astype("int")
            rects.append(person_box)

    boundingboxes = np.array(rects)
    boundingboxes = boundingboxes.astype(int)
    rects = non_max_suppression_fast(boundingboxes, 0.3)
```

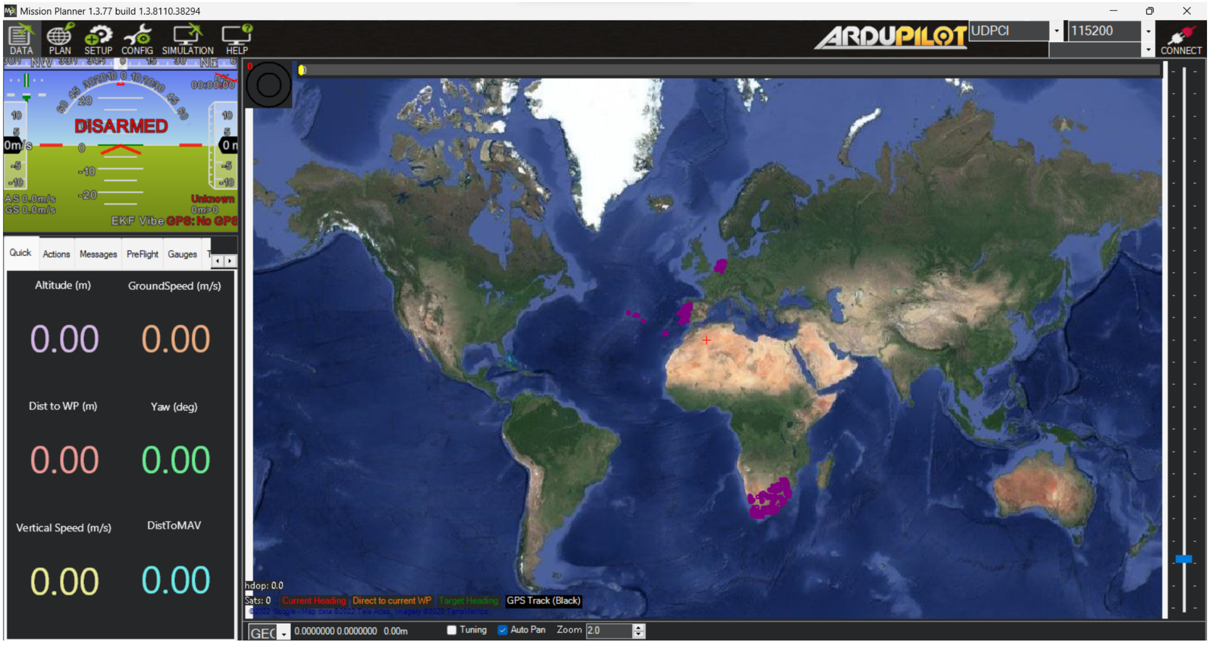
Defining an ID number for every user detected

```python
# import the necessary packages
from scipy.spatial import distance as dist
from collections import OrderedDict
import numpy as np


class CentroidTracker:
    def __init__(self, maxDisappeared=50, maxDistance=50):
        self.nextObjectID = 0
        self.objects = OrderedDict()
        self.disappeared = OrderedDict()
        self.bbox = OrderedDict()  # CHANG
        self.maxDisappeared = maxDisappeared

        self.maxDistance = maxDistance

    def register(self, centroid, inputRect):
        self.objects[self.nextObjectID] = centroid
        self.bbox[self.nextObjectID] = inputRect  # CHANGE
        self.disappeared[self.nextObjectID] = 0
        self.nextObjectID += 1

    def deregister(self, objectID):
        del self.objects[objectID]
        del self.disappeared[objectID]
        del self.bbox[objectID]  # CHANGE

    def update(self, rects):
        if len(rects) == 0:
            for objectID in list(self.disappeared.keys()):
                self.disappeared[objectID] += 1

                if self.disappeared[objectID] > self.maxDisappeared:
                    self.deregister(objectID)
```

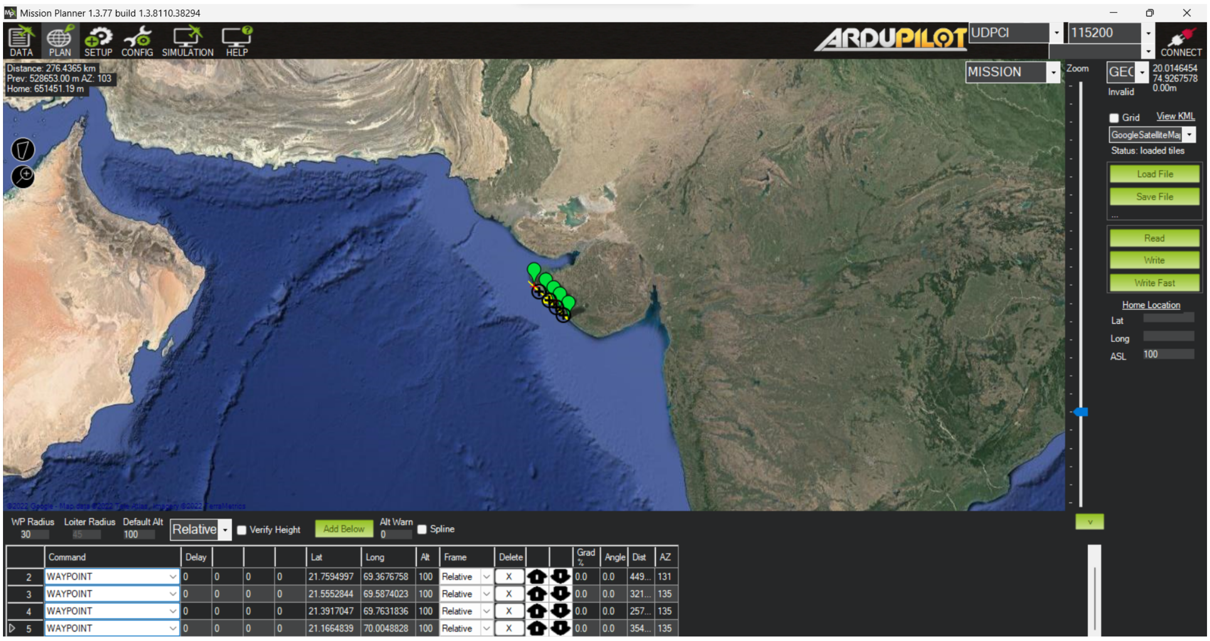
Displaying the count and total number on the frame screen

```python
lpc_count = len(objects)
opc_count = len(object_id_list)

lpc_txt = "People in frame: {}".format(lpc_count)
opc_txt = "Total people count: {}".format(opc_count)

cv2.putText(frame, lpc_txt, (5, 60), cv2.FONT_HERSHEY_COMPLEX_SMALL, 1, (0, 0, 255), 1)
cv2.putText(frame, opc_txt, (5, 90), cv2.FONT_HERSHEY_COMPLEX_SMALL, 1, (0, 0, 255), 1)

cv2.imshow("Application", frame)
key = cv2.waitKey(1)
if key == ord('q'):
    break
```

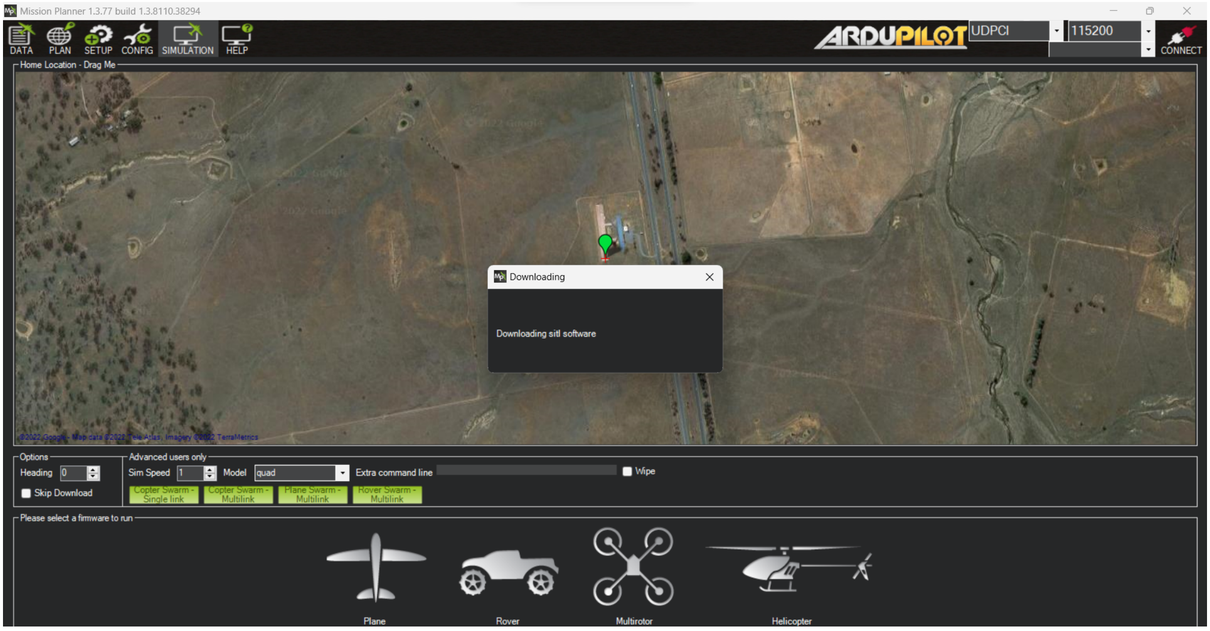
Mission planner



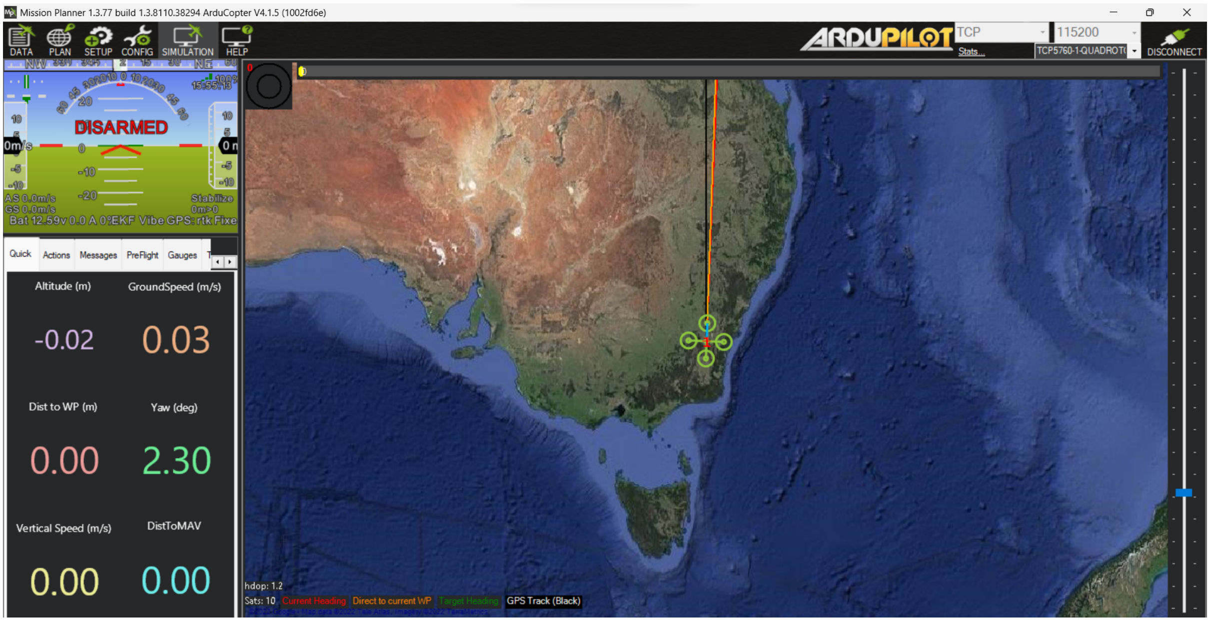
Defining the path across the border



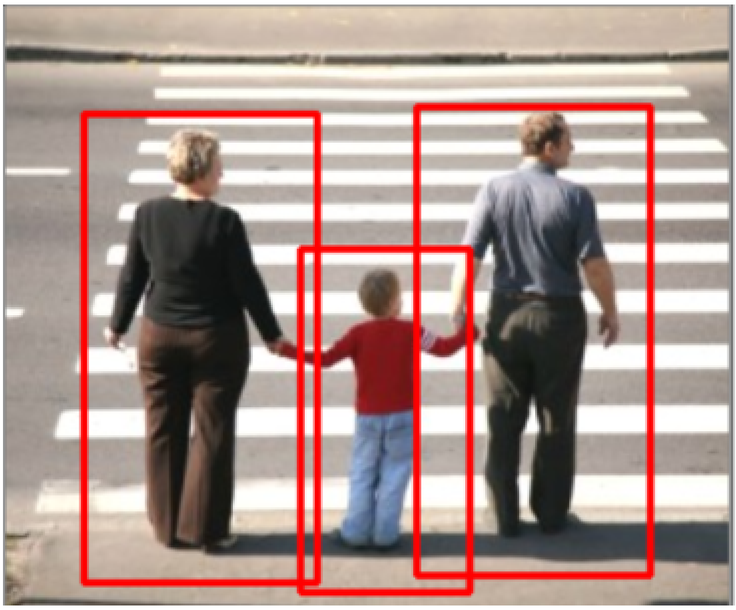
Downloading the SITL software before executing the code



Drone is disarmed and working



Output



Link to the demonstration video:-
https://drive.google.com/drive/folders/1xeY7vLsrOaPfPg3OePgIgZwTxHHHIfRb?usp=sharing

REFERENCES

- All figures in this report have been made using www.creately.com
- https://robu.in/product/emax-mt2213-935kv-bldc-motor-ccw-prop1045-combo-original/?gclid=EAIaIQobChMIyvDo34Wb7AIV1H0rCh3ddAgtEAQYASABEgKc5vD_BwE
- https://robu.in/product/orange-hd-propellers-104510x4-5-abs-dji-black-1cw1ccw-1pair-premium-quality/?gclid=EAIaIQobChMI5o7a44ab7AIVgXwrCh3fkAmcEAQYAiABEgJJmPD_BwE
- https://robu.in/product/orange-5200mah-3s-40c80c-lithium-polymer-battery-pack-lipo/?gclid=EAIaIQobChMIvdXlI4ib7AIVDH8rCh0gUQhIEAQYAyABEgLCOfD_BwE
- https://www.amazon.com/AKK-BLHeli_S-Electronic-Controller-Racing/dp/B079FSFGNX
- https://robu.in/product/apm-2-5-2-apm-2-6-pixhawk-power-module-v1-0-output-bec-3a-xt60-plug-28v-90a/?gclid=EAIaIQobChMI2NvXtYeb7AIVViUrCh3zBwEtEAQYASABEgL6mfD_BwE
- https://ardupilot.org/dev/docs/raspberry-pi-via-mavlink.html
- https://pjreddie.com/darknet/yolo/
- https://dronekit.io/
- West, J. P., & Bowman, J. S. (2016). The domestic use of drones: An ethical analysis of surveillance issues. *Public Administration Review*, *76*(4), 649-659.
- The Supreme Court remarked in Kyllo v. United States, 533 U.S. 27, 37 (2001):
- U.S. Supports Some Domestic Drone Use, But Public Registers Concern About Own Privacy 1, Monmouth University Polling Institute (June 12, 2012), available at http://www.monmouth.edu.
- Thompson, R. M. (2012, September). Drones in domestic surveillance operations: Fourth amendment implications and legislative responses. Congressional Research Service, Library of Congress.Mohammad Aazam, Sherali Zeadally, and Khaled A. Harras. 2018b. Fog computing architecture, evaluation, and future research directions. IEEE Communications Magazine 56, 5 (2018), 46--52.
- Mohammad Aazam, Sherali Zeadally, and Khaled A. Harras. 2018a. Offloading in fog computing for IoT: Review, enabling technologies, and research opportunities. Future Generation Computer Systems 87 (2018), 278--289.
- Ahmed Abdelkader, Moamen Mokhtar, and Hazem El-Alfy. 2012. Angular heuristics for coverage maximization in multi-camera surveillance. In AVSS'12.
- Tseng, C. M., Chau, C. K., Elbassioni, K. M., & Khonji, M. (2017). Flight tour planning with recharging optimization for battery-operated autonomous drones. *CoRR, abs/1703.10049*.