IoT Connectivity for Drone Applications in Smart Agriculture

Guide: Dr. Y Raja Vara Prasad
B22RVP03

Team members:

K Litheesh Kumar - S20190020218

Surya Sathvik -

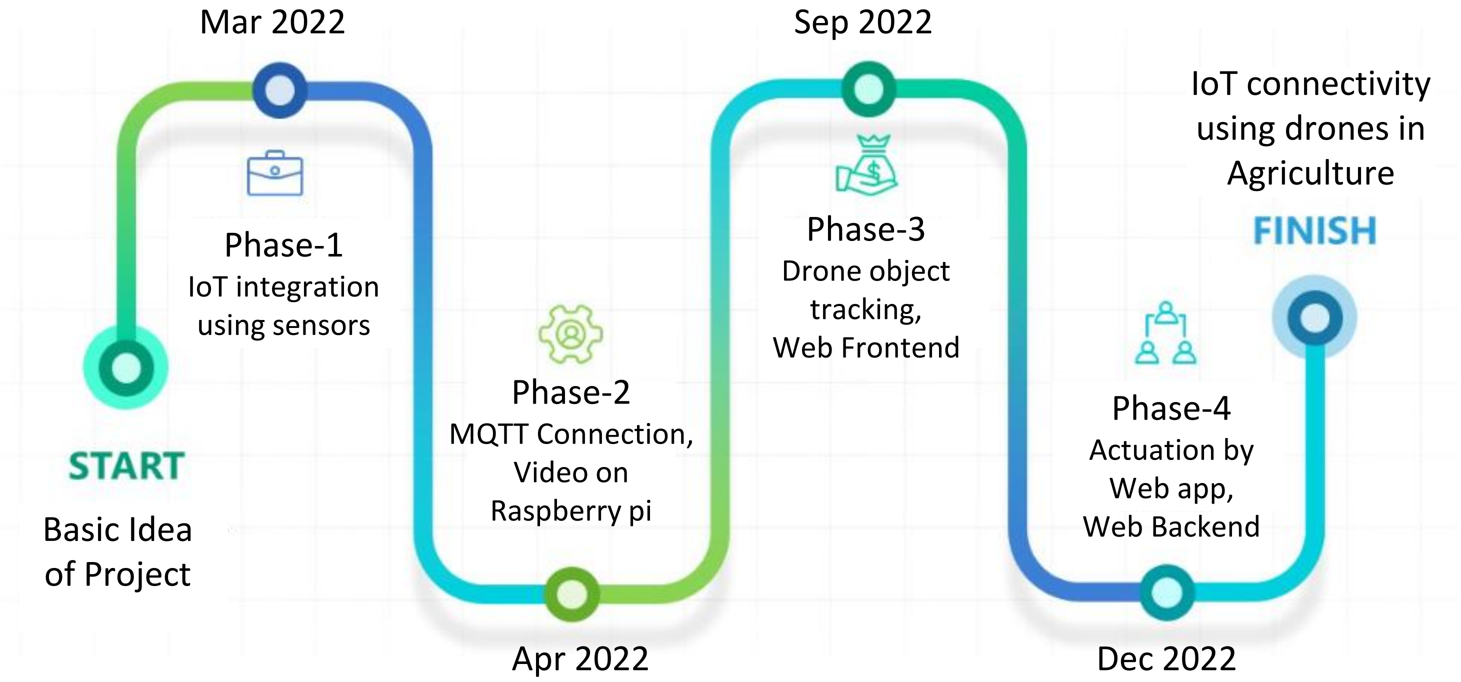
Problem Statement

The connectivity limitations of smart agriculture and its solutions are analyzed in this project.

- Writing python code for automating drone path through computer vision

- We are creating a user-friendly website from which users can easily access and control processes

Timeline



START

Basic Idea
of Project

Mar 2022

Phase-1
IoT integration
using sensors

Apr 2022

Phase-2
MQTT Connection,
Video on
Raspberry pi

Sep 2022

Phase-3
Drone object
tracking,
Web Frontend

Dec 2022

Phase-4
Actuation by
Web app,
Web Backend

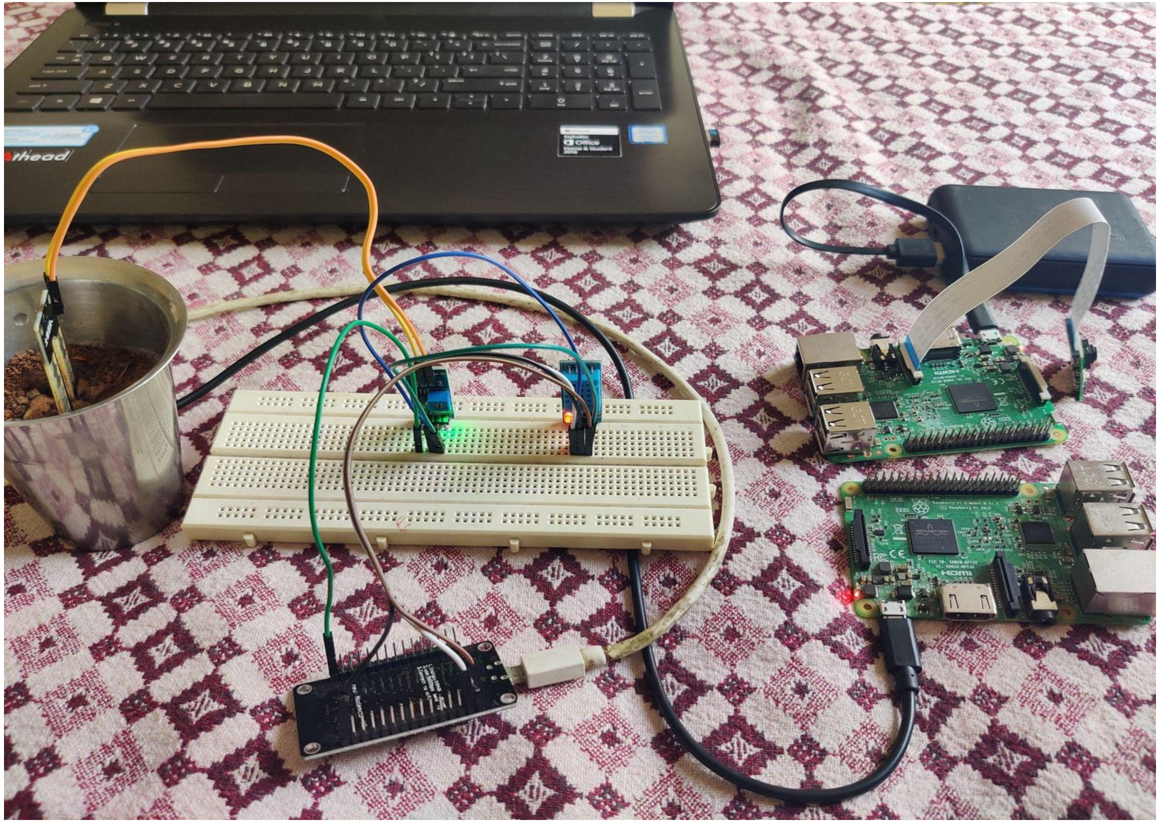
FINISH

IoT connectivity
using drones in
Agriculture

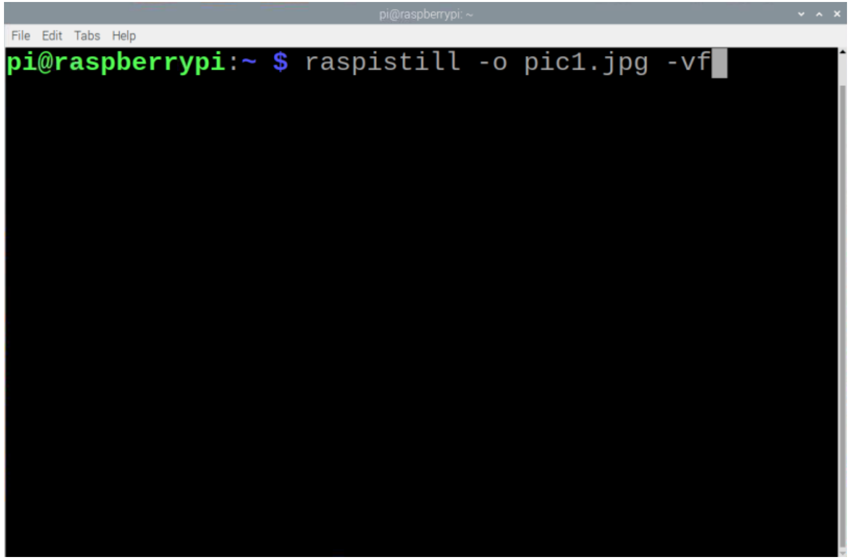
Recap: Phase1, 2 work

- Sensors are connected to Raspberry pi through NodeMCU.
- Pump Motor is connected to another NodeMCU.
- Both NodeMCUs are connected to Ground Raspberry pi using MQTT.
- Ground Raspberry pi changes to client from broker
- Drone Raspberry pi acts as MQTT broker
- and published sensor data on MQTT Cloud (Adafruit).
- Fixing a camera module to Drone Raspberry
- Analyzed the delay in streaming in various resolutions.

Hardware Connections

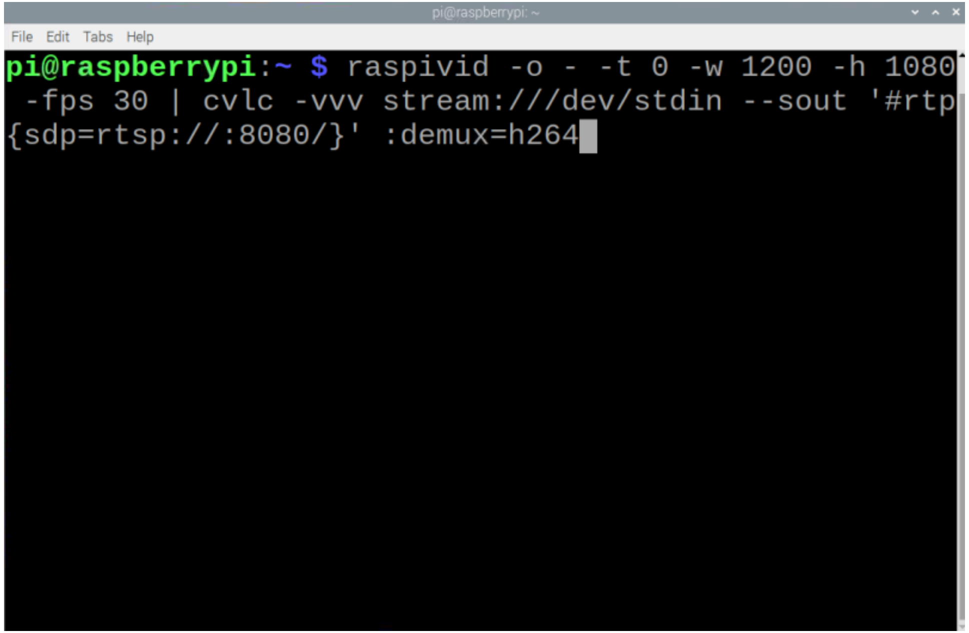


Raspberry pi Video Streaming



```
pi@raspberrypi:~ $ raspistill -o pic1.jpg -vf
```

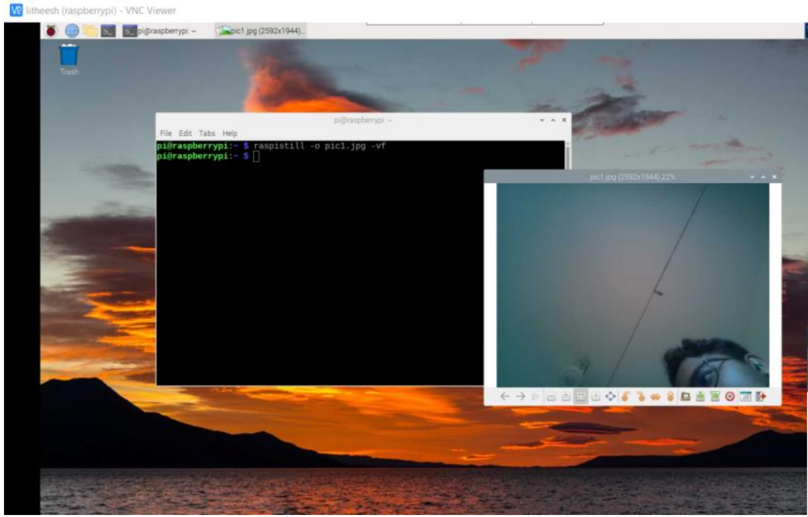
Command line for taking photos from drone raspberry pi



```
pi@raspberrypi:~ $ raspivid -o - -t 0 -w 1200 -h 1080
 -fps 30 | cvlc -vvv stream:///dev/stdin --sout '#rtp
{sdp=rtsp://:8080/}' :demux=h264
```

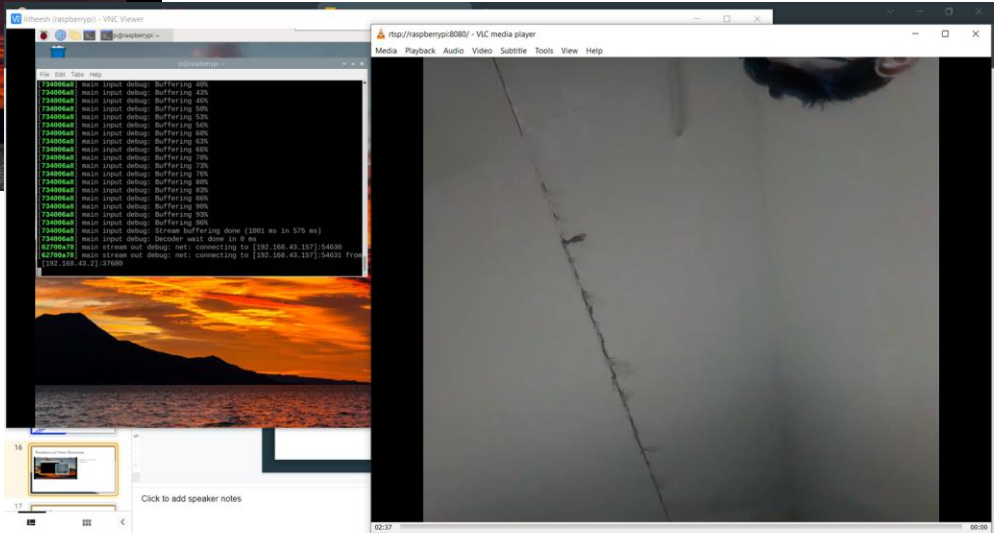
Command line for Live streaming video from drone raspberry pi

Video & Photo



```
pi@raspberrypi:~ $ raspistill -o pic1.jpg -vf
pi@raspberrypi:~ $ 
```

Taking photos from drone raspberry pi


litheesh (raspberrypi) - VNC Viewer
pi@raspberrypi: ~
pi@raspberrypi: ~
File Edit Tabs Help
rtsp://raspberrypi:8080/ - VLC media player
Media Playback Audio Video Subtitle Tools View Help
02:37
00:00
16
17
Click to add speaker notes

```
[734006a8] main input debug: Buffering 40%
[734006a8] main input debug: Buffering 43%
[734006a8] main input debug: Buffering 46%
[734006a8] main input debug: Buffering 50%
[734006a8] main input debug: Buffering 53%
[734006a8] main input debug: Buffering 56%
[734006a8] main input debug: Buffering 60%
[734006a8] main input debug: Buffering 63%
[734006a8] main input debug: Buffering 66%
[734006a8] main input debug: Buffering 70%
[734006a8] main input debug: Buffering 73%
[734006a8] main input debug: Buffering 76%
[734006a8] main input debug: Buffering 80%
[734006a8] main input debug: Buffering 83%
[734006a8] main input debug: Buffering 86%
[734006a8] main input debug: Buffering 90%
[734006a8] main input debug: Buffering 93%
[734006a8] main input debug: Buffering 96%
[734006a8] main input debug: Stream buffering done (1001 ms in 575 ms)
[734006a8] main input debug: Decoder wait done in 0 ms
[62700a78] main stream out debug: net: connecting to [192.168.43.157]:54630
[62700a78] main stream out debug: net: connecting to [192.168.43.157]:54631 from
 [192.168.43.2]:37680
```

Live streaming video from drone raspberry pi

Phase3 work

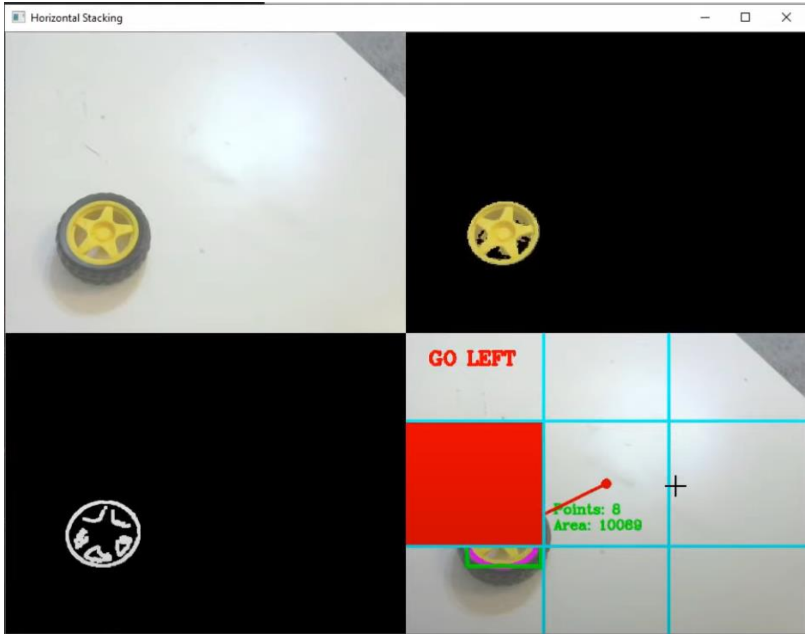
- We are using Tello drone
- It can be flown both through app and commands
- Python code for automated flying mode
- It detects the object by Computer Vision(CV)
- follows the objects and reaches the nearest path to the object
- And take photos and videos at that particular location.
- Front End of the Website.

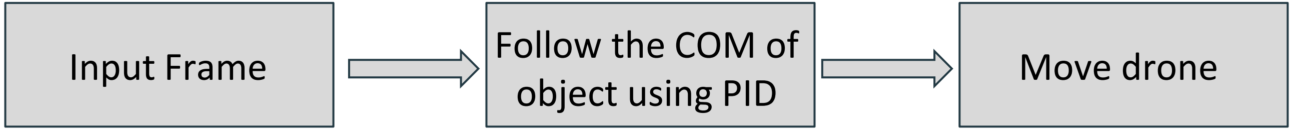
Why Tello?

- Tello is a small quadcopter
- features a Vision position system
- an onboard camera.
- Using its Vision Positioning System and advanced flight controller,
- it can hover in place and is suitable for flying indoors.
- Advanced features like Bounce mode, 8D Flips, and EZ Shots make using Tello fun.

Object Tracking Drone

- Identify different objects in the captured image based on the color
- image to an HSI plane (HUE, SATURATION, and INTENSITY)
- We get the binary mask of our object
- try to put the COM of the object in the frame
- Use PID to calculate the speed and movements of the drone

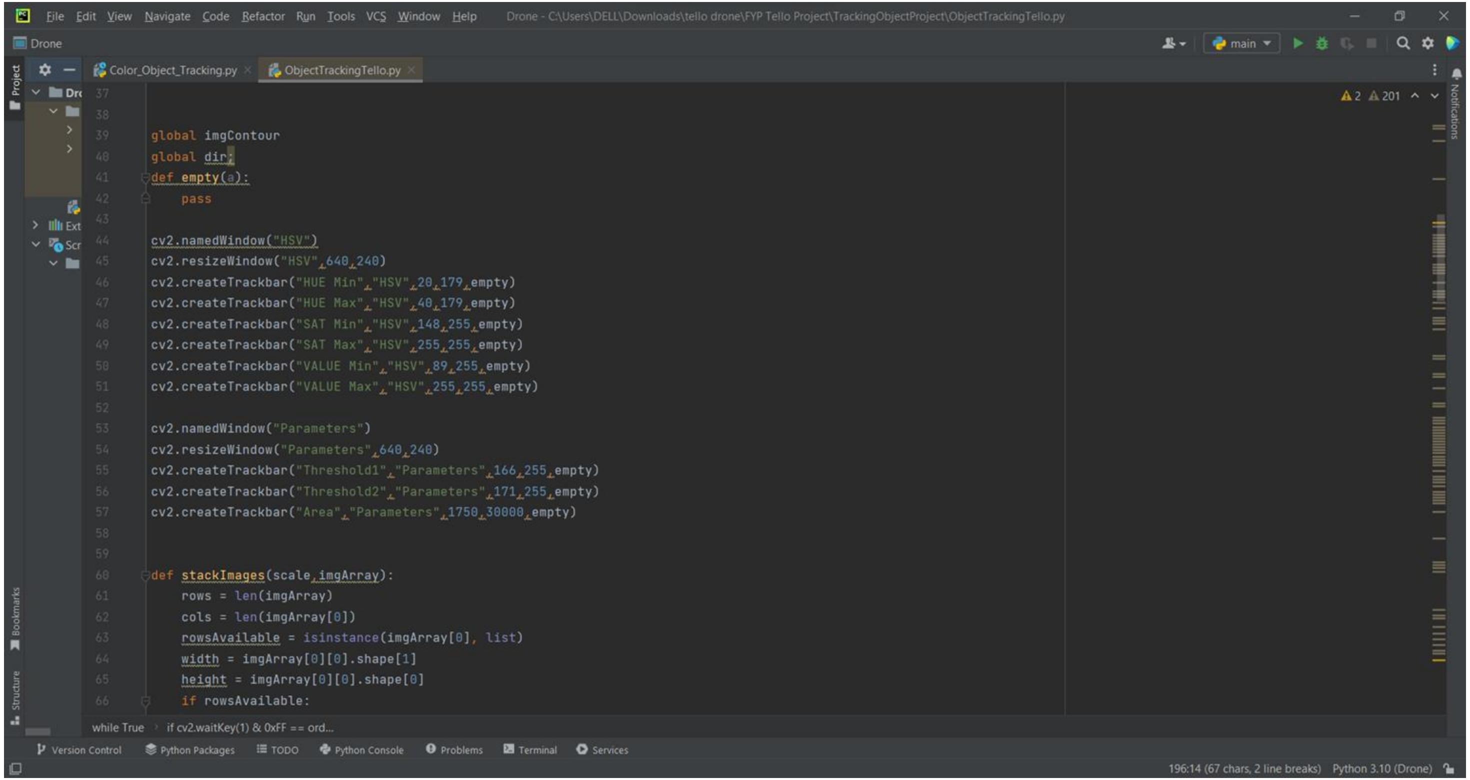



Input Frame → Follow the COM of object using PID → Move drone

Technology Challenges:

- Working with hardware has a set of challenges that running simulations does not provide.
- The issues of dying batteries and overheating.
- The DJI Tello will not take flight if it senses that it is overheating.
- Carpeted floors have been observed to prevent from cooling down.
- Finally, several libraries must be downloaded to run Tello with CV2.

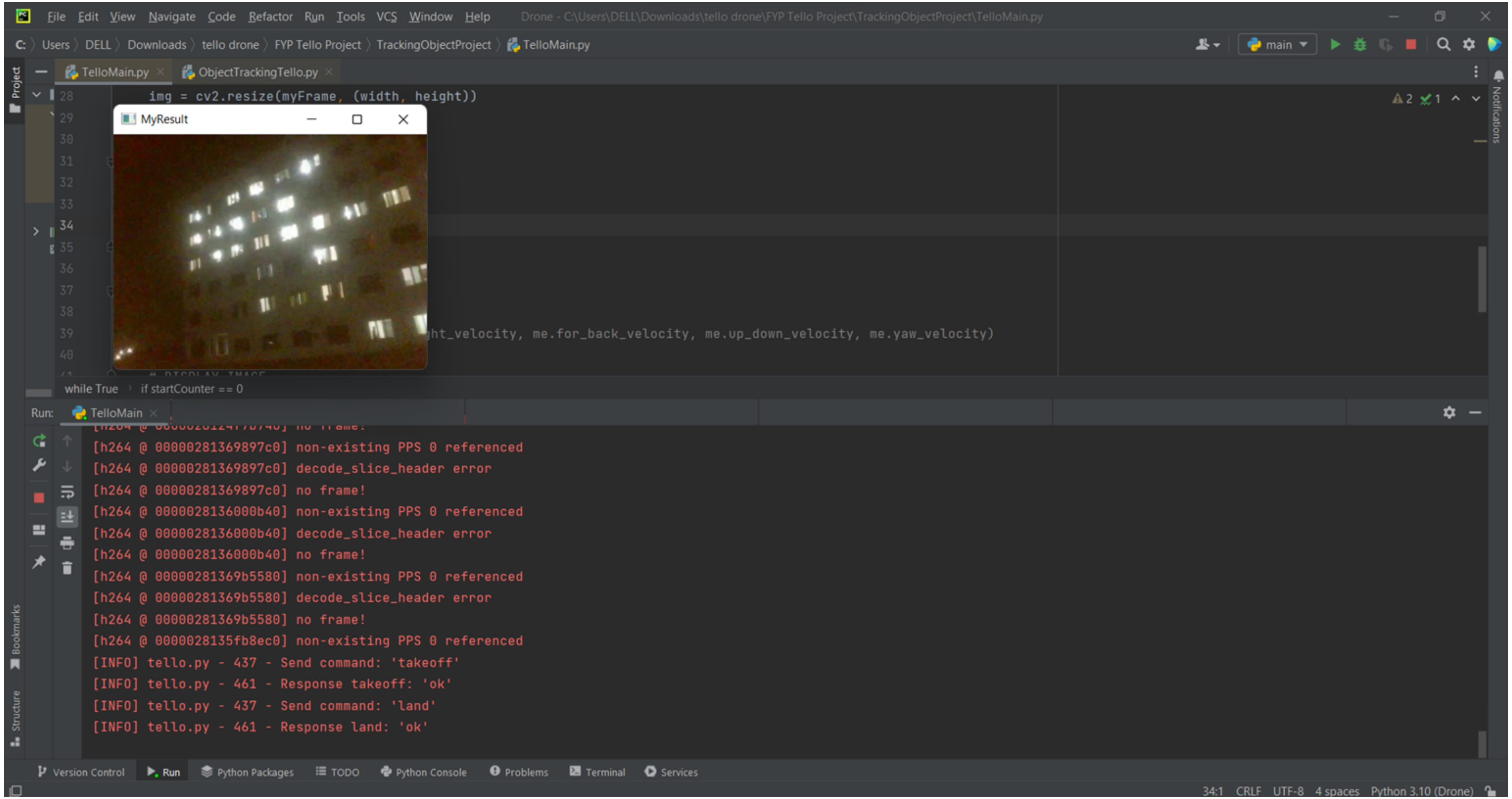
Python code:


File Edit View Navigate Code Refactor Run Tools VCS Window Help
Drone - C:\Users\DELL\Downloads\tello drone\FYP Tello Project\TrackingObjectProject\ObjectTrackingTello.py
Drone
main
Color_Object_Tracking.py
ObjectTrackingTello.py
Project
Dro
Ext
Scr
Bookmarks
Structure
Notifications
2 201
while True
if cv2.waitKey(1) & 0xFF == ord...
Version Control
Python Packages
TODO
Python Console
Problems
Terminal
Services
196:14 (67 chars, 2 line breaks)
Python 3.10 (Drone)

```python


global imgContour
global dir;
def empty(a):
    pass

cv2.namedWindow("HSV")
cv2.resizeWindow("HSV",640,240)
cv2.createTrackbar("HUE Min","HSV",20,179,empty)
cv2.createTrackbar("HUE Max","HSV",40,179,empty)
cv2.createTrackbar("SAT Min","HSV",148,255,empty)
cv2.createTrackbar("SAT Max","HSV",255,255,empty)
cv2.createTrackbar("VALUE Min","HSV",89,255,empty)
cv2.createTrackbar("VALUE Max","HSV",255,255,empty)

cv2.namedWindow("Parameters")
cv2.resizeWindow("Parameters",640,240)
cv2.createTrackbar("Threshold1","Parameters",166,255,empty)
cv2.createTrackbar("Threshold2","Parameters",171,255,empty)
cv2.createTrackbar("Area","Parameters",1750,30000,empty)


def stackImages(scale,imgArray):
    rows = len(imgArray)
    cols = len(imgArray[0])
    rowsAvailable = isinstance(imgArray[0], list)
    width = imgArray[0][0].shape[1]
    height = imgArray[0][0].shape[0]
    if rowsAvailable:
```

Link for Python code: https://github.com/litheeshkumar/

Python output:

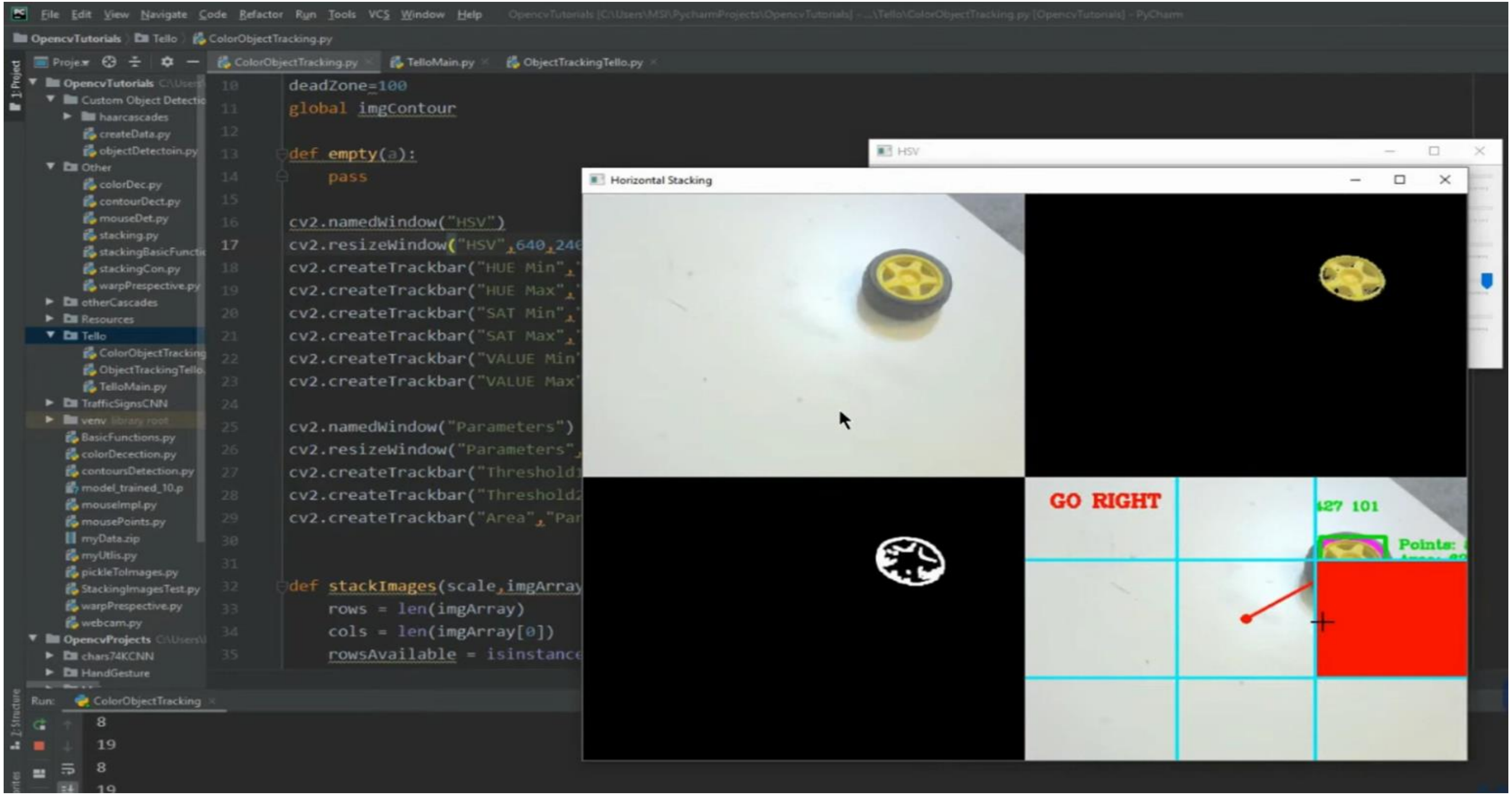

File Edit View Navigate Code Refactor Run Tools VCS Window Help
Drone - C:\Users\DELL\Downloads\tello drone\FYP Tello Project\TrackingObjectProject\TelloMain.py
C: 〉 Users 〉 DELL 〉 Downloads 〉 tello drone 〉 FYP Tello Project 〉 TrackingObjectProject 〉 TelloMain.py
main
TelloMain.py
ObjectTrackingTello.py
Project
MyResult
Notifications
while True 〉 if startCounter == 0
Run: TelloMain
Bookmarks
Structure
Version Control
Run
Python Packages
TODO
Python Console
Problems
Terminal
Services
34:1 CRLF UTF-8 4 spaces Python 3.10 (Drone)

```
    img = cv2.resize(myFrame, (width, height))

ght_velocity, me.for_back_velocity, me.up_down_velocity, me.yaw_velocity)
```

```
[h264 @ 00000281369897c0] non-existing PPS 0 referenced
[h264 @ 00000281369897c0] decode_slice_header error
[h264 @ 00000281369897c0] no frame!
[h264 @ 0000028136000b40] non-existing PPS 0 referenced
[h264 @ 0000028136000b40] decode_slice_header error
[h264 @ 0000028136000b40] no frame!
[h264 @ 00000281369b5580] non-existing PPS 0 referenced
[h264 @ 00000281369b5580] decode_slice_header error
[h264 @ 00000281369b5580] no frame!
[h264 @ 0000028135fb8ec0] non-existing PPS 0 referenced
[INFO] tello.py - 437 - Send command: 'takeoff'
[INFO] tello.py - 461 - Response takeoff: 'ok'
[INFO] tello.py - 437 - Send command: 'land'
[INFO] tello.py - 461 - Response land: 'ok'
```

Results

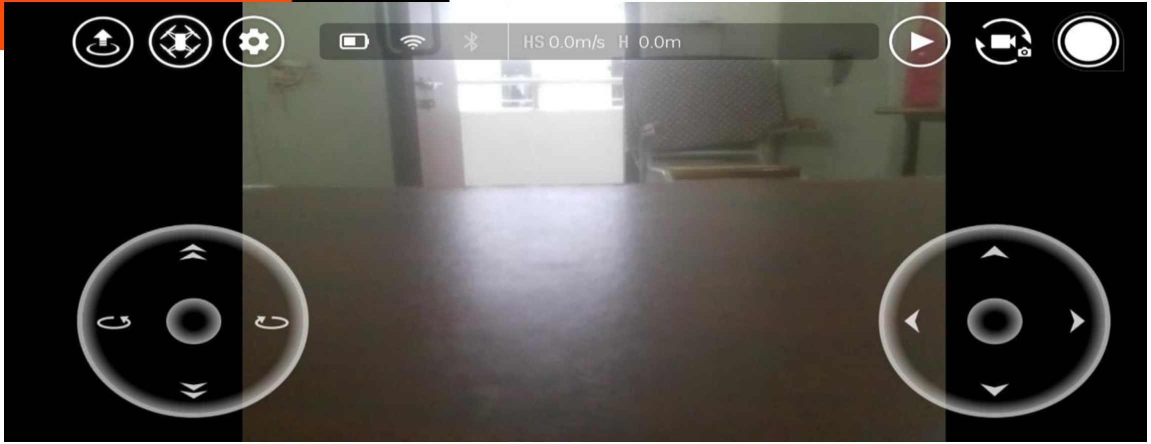
Results



```
deadZone=100
global imgContour

def empty(a):
    pass

cv2.namedWindow("HSV")
cv2.resizeWindow("HSV",640,240
cv2.createTrackbar("HUE Min",
cv2.createTrackbar("HUE Max",
cv2.createTrackbar("SAT Min",
cv2.createTrackbar("SAT Max",
cv2.createTrackbar("VALUE Min"
cv2.createTrackbar("VALUE Max"

cv2.namedWindow("Parameters")
cv2.resizeWindow("Parameters",
cv2.createTrackbar("Threshold1
cv2.createTrackbar("Threshold2
cv2.createTrackbar("Area","Par


def stackImages(scale,imgArray
    rows = len(imgArray)
    cols = len(imgArray[0])
    rowsAvailable = isinstance
```

```
8
19
8
19
```

Results - Tello app photos



Video links

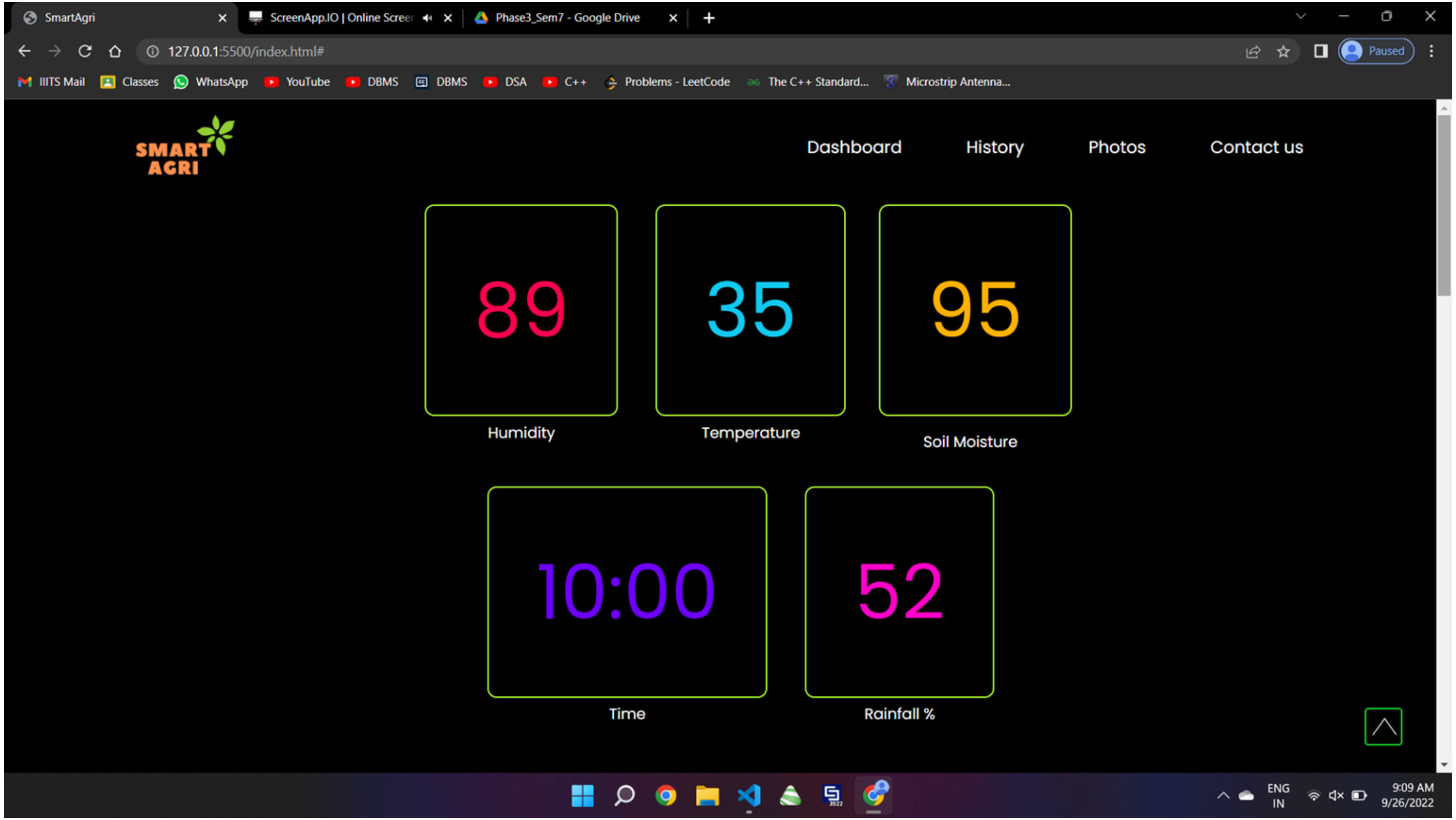
Here is the video link of the Phase 2 Work:

https://drive.google.com/file/d/1GPMdVgjzUaGgT7chgbvmTJjrJtJr4eWn/view?usp=sharing

Google drive link for pics and video taken by Tello:

https://drive.google.com/drive/folders/1_X5nZUt6YE7Y_w4c28JQN6mlXbONluxs?usp=sharing

Website Pics



89
Humidity

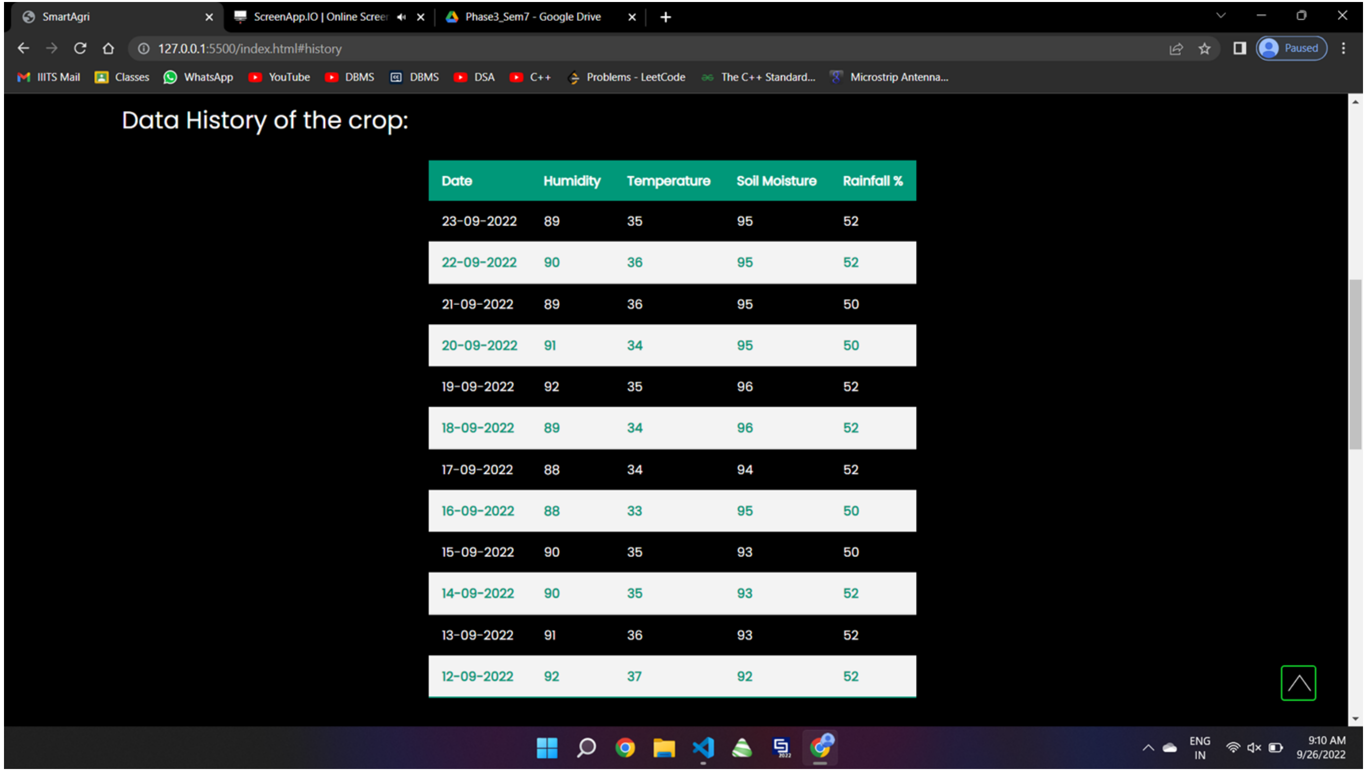
35
Temperature

95
Soil Moisture

10:00
Time

52
Rainfall %

Website Pics



Data History of the crop:

Date	Humidity	Temperature	Soil Moisture	Rainfall %
23-09-2022	89	35	95	52
22-09-2022	90	36	95	52
21-09-2022	89	36	95	50
20-09-2022	91	34	95	50
19-09-2022	92	35	96	52
18-09-2022	89	34	96	52
17-09-2022	88	34	94	52
16-09-2022	88	33	95	50
15-09-2022	90	35	93	50
14-09-2022	90	35	93	52
13-09-2022	91	36	93	52
12-09-2022	92	37	92	52

Link for Website code: https://github.com/litheeshkumar/

Expected Outcomes

PHASE-4: December 2022

1. Optimizing CV (Computer Vision) model for object tracking
2. Coding for giving instructions from Website to ground level Actuators like pump motor etc..
3. Website Backend

References

1. Jeffrey Kantor. CBE 30338 Chemical Process Control. IEEE Access 2019, 7, 129551–129583. [CrossRef]

1. A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. International Journal of Robotics Research (IJRR), 2013. [CrossRef]

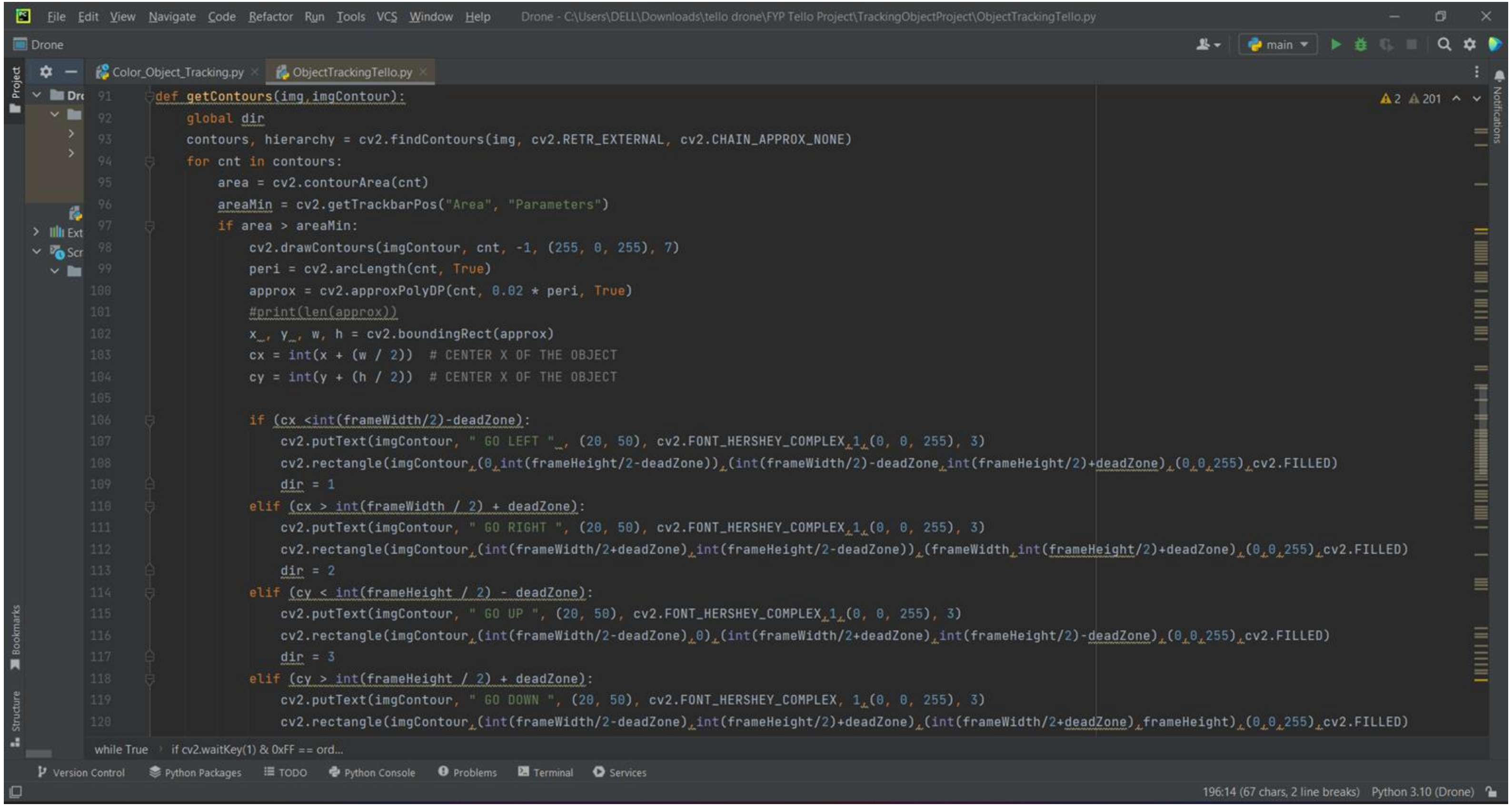
1. R. Girshick. Fast r-cnn. In International Conference on Computer Vision (ICCV), 2015. [CrossRef]

1. S. Han, H. Mao, and W. J. Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. arXiv preprint arXiv:1510.00149, 2015. [CrossRef]

THANK YOU

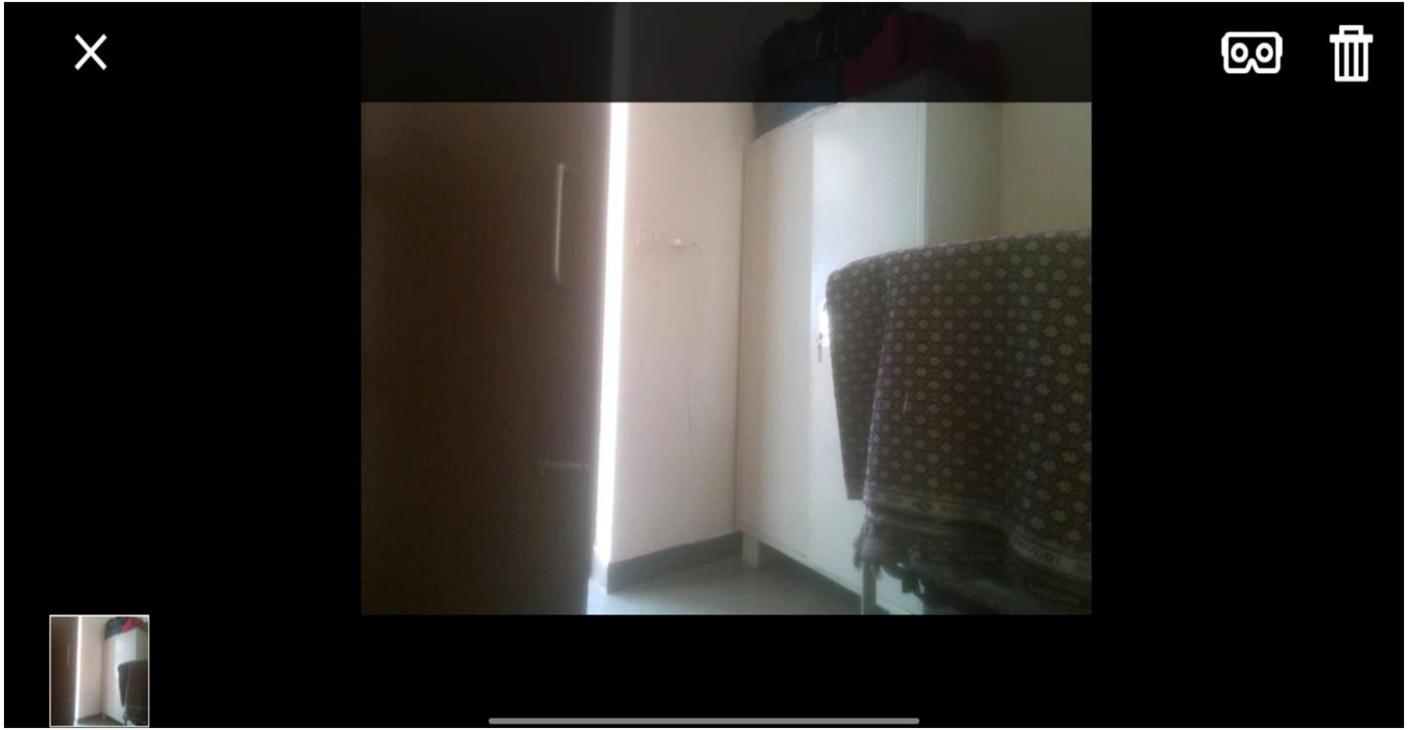
Drone pics:

Python code:

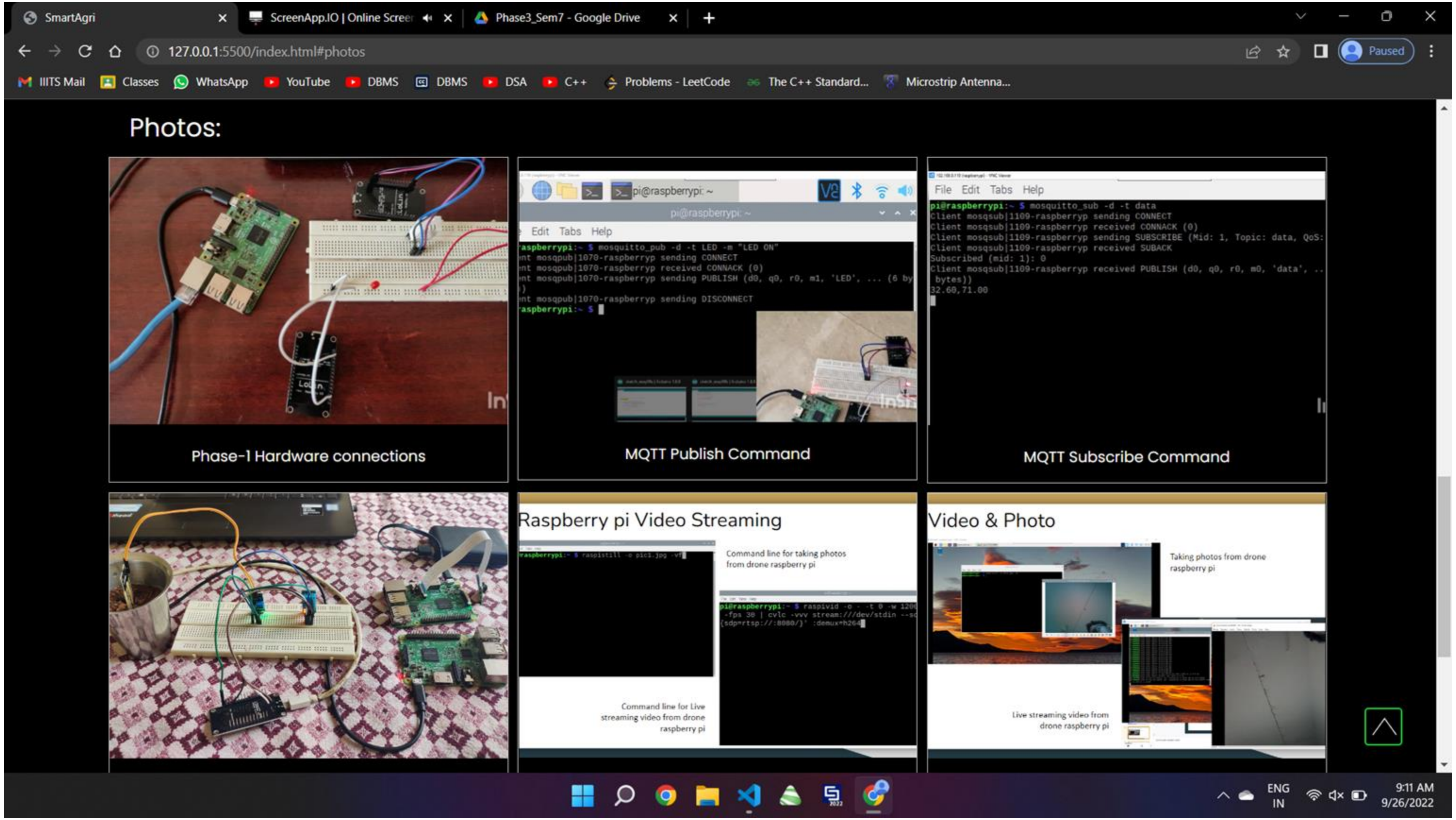

File Edit View Navigate Code Refactor Run Tools VCS Window Help
Drone - C:\Users\DELL\Downloads\tello drone\FYP Tello Project\TrackingObjectProject\ObjectTrackingTello.py
Drone
main
Project
Color_Object_Tracking.py
ObjectTrackingTello.py
Notifications
Bookmarks
Structure
while True
if cv2.waitKey(1) & 0xFF == ord...
Version Control
Python Packages
TODO
Python Console
Problems
Terminal
Services
196:14 (67 chars, 2 line breaks)
Python 3.10 (Drone)

```python
def getContours(img,imgContour):
    global dir
    contours, hierarchy = cv2.findContours(img, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_NONE)
    for cnt in contours:
        area = cv2.contourArea(cnt)
        areaMin = cv2.getTrackbarPos("Area", "Parameters")
        if area > areaMin:
            cv2.drawContours(imgContour, cnt, -1, (255, 0, 255), 7)
            peri = cv2.arcLength(cnt, True)
            approx = cv2.approxPolyDP(cnt, 0.02 * peri, True)
            #print(len(approx))
            x, y, w, h = cv2.boundingRect(approx)
            cx = int(x + (w / 2))  # CENTER X OF THE OBJECT
            cy = int(y + (h / 2))  # CENTER X OF THE OBJECT

            if (cx <int(frameWidth/2)-deadZone):
                cv2.putText(imgContour, " GO LEFT ", (20, 50), cv2.FONT_HERSHEY_COMPLEX,1,(0, 0, 255), 3)
                cv2.rectangle(imgContour,(0,int(frameHeight/2-deadZone)),(int(frameWidth/2)-deadZone,int(frameHeight/2)+deadZone),(0,0,255),cv2.FILLED)
                dir = 1
            elif (cx > int(frameWidth / 2) + deadZone):
                cv2.putText(imgContour, " GO RIGHT ", (20, 50), cv2.FONT_HERSHEY_COMPLEX,1,(0, 0, 255), 3)
                cv2.rectangle(imgContour,(int(frameWidth/2+deadZone),int(frameHeight/2-deadZone)),(frameWidth,int(frameHeight/2)+deadZone),(0,0,255),cv2.FILLED)
                dir = 2
            elif (cy < int(frameHeight / 2) - deadZone):
                cv2.putText(imgContour, " GO UP ", (20, 50), cv2.FONT_HERSHEY_COMPLEX,1,(0, 0, 255), 3)
                cv2.rectangle(imgContour,(int(frameWidth/2-deadZone),0),(int(frameWidth/2+deadZone),int(frameHeight/2)-deadZone),(0,0,255),cv2.FILLED)
                dir = 3
            elif (cy > int(frameHeight / 2) + deadZone):
                cv2.putText(imgContour, " GO DOWN ", (20, 50), cv2.FONT_HERSHEY_COMPLEX, 1,(0, 0, 255), 3)
                cv2.rectangle(imgContour,(int(frameWidth/2-deadZone),int(frameHeight/2)+deadZone),(int(frameWidth/2+deadZone),frameHeight),(0,0,255),cv2.FILLED)
```

Photos taken by drone:



Website Pics



Photos:

Phase-1 Hardware connections

```
aspberrypi:~ $ mosquitto_pub -d -t LED -m "LED ON"
ent mosqpub|1070-raspberryp sending CONNECT
ent mosqpub|1070-raspberryp received CONNACK (0)
ent mosqpub|1070-raspberryp sending PUBLISH (d0, q0, r0, m1, 'LED', ... (6 by
)
ent mosqpub|1070-raspberryp sending DISCONNECT
aspberrypi:~ $
```

MQTT Publish Command

```
pi@raspberrypi:~ $ mosquitto_sub -d -t data
Client mosqsub|1109-raspberryp sending CONNECT
Client mosqsub|1109-raspberryp received CONNACK (0)
Client mosqsub|1109-raspberryp sending SUBSCRIBE (Mid: 1, Topic: data, QoS:
Client mosqsub|1109-raspberryp received SUBACK
Subscribed (mid: 1): 0
Client mosqsub|1109-raspberryp received PUBLISH (d0, q0, r0, m0, 'data', ..
 bytes))
32.60,71.00
```

MQTT Subscribe Command

Raspberry pi Video Streaming

```
raspberrypi:~ $ raspistill -o pic1.jpg -vf
```

Command line for taking photos from drone raspberry pi

```
pi@raspberrypi:~ $ raspivid -o - -t 0 -w 120
 -fps 30 | cvlc -vvv stream:///dev/stdin --s
{sdp=rtsp://:8080/}' :demux=h264
```

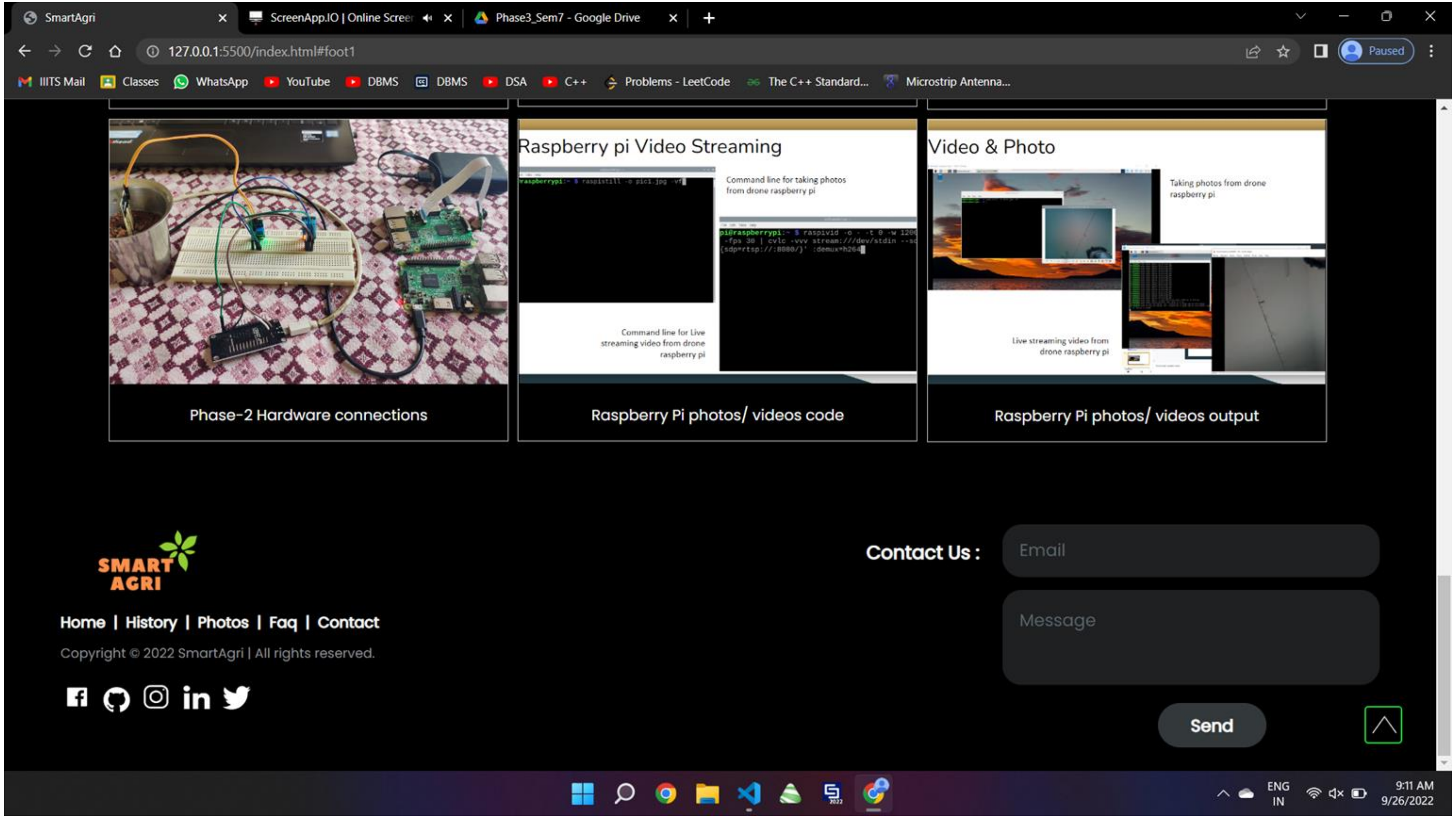
Command line for Live streaming video from drone raspberry pi

Video & Photo

Taking photos from drone raspberry pi

Live streaming video from drone raspberry pi

Website Pics



Phase-2 Hardware connections

Raspberry pi Video Streaming

Command line for taking photos from drone raspberry pi

```
raspberrypi:~ $ raspistill -o pic1.jpg -vf
```

Command line for Live streaming video from drone raspberry pi

```
pi@raspberrypi:~ $ raspivid -o - -t 0 -w 1200
 -fps 30 | cvlc -vvv stream:///dev/stdin --sd
{sdp=rtsp://:8080/}' :demux=h264
```

Raspberry Pi photos/ videos code

Video & Photo

Taking photos from drone raspberry pi

Live streaming video from drone raspberry pi

Raspberry Pi photos/ videos output

Home | History | Photos | Faq | Contact

Copyright © 2022 SmartAgri | All rights reserved.

Contact Us :

Email

Message

Send