



Lesson 6:

Data gathering with Drones

Smart Farming with Drones

Drones4STEM © 2023 by Drones4STEM is licensed under CC BY-NC-SA 4.0. To view a copy of this license, visit <https://creativecommons.org/licenses/by-nc-sa/4.0/>

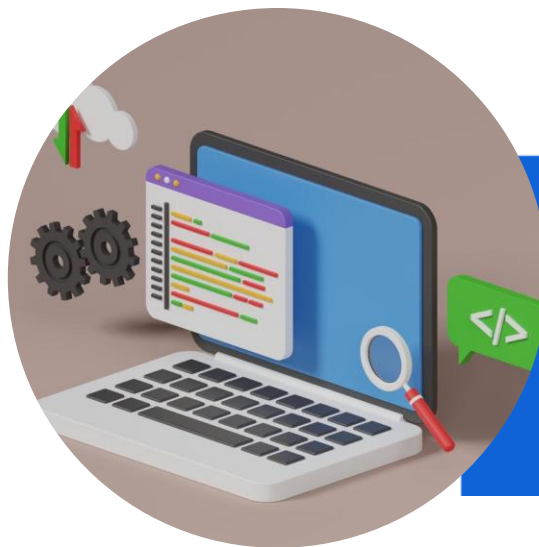


Co-funded by
the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

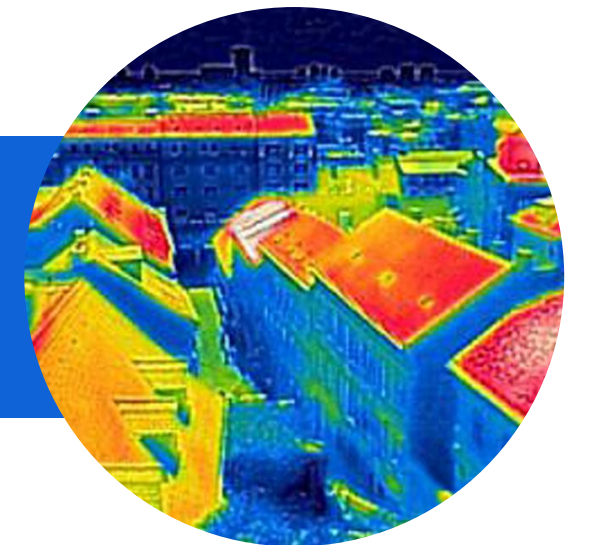
Lesson Objectives

Understand how drones gather data
in agriculture



Learn how to use python to take a picture

See how aerial imaging helps farmers monitor their fields



TOPICS

What is data gathering in Smart Farming

Sensors, Aerial Images and automation

Drones as data collectors

Discussion and Q&A

What is data gathering in Smart Farming



Smart farming uses advanced technologies like drones and sensors to collect real-time data from the field. This data helps farmers make better decisions about crop health, irrigation, fertilization, and pest control.

- Drones act as "flying scouts" that observe large fields quickly.
- Cameras and sensors on drones collect images and environmental data.
- Data gathering helps detect problems early—before they spread.

Mission Purpose

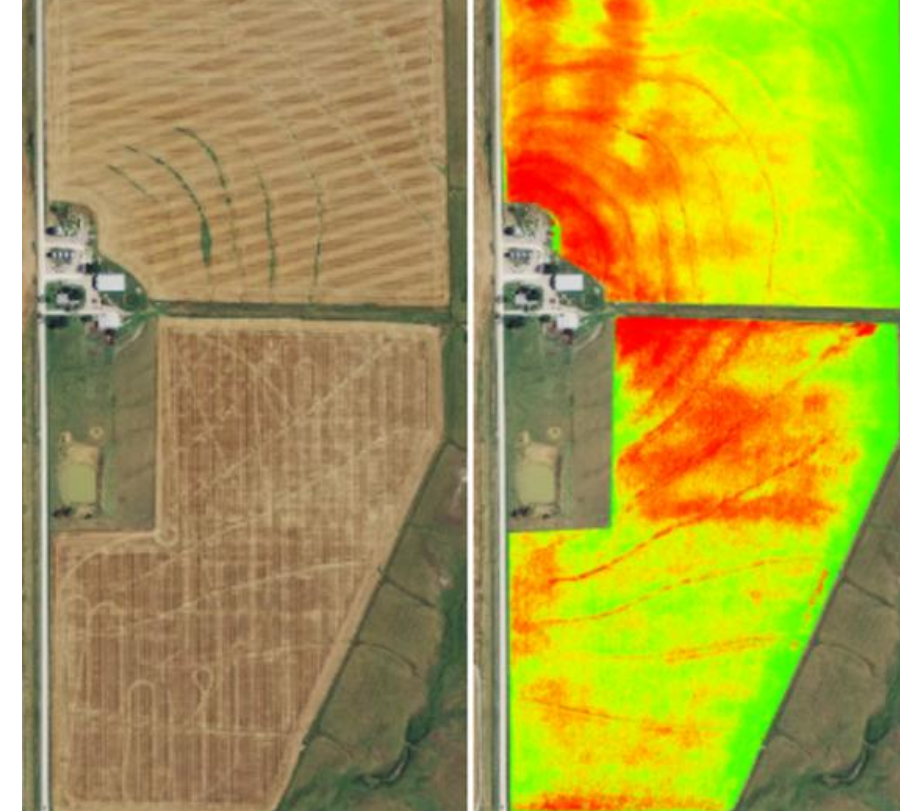
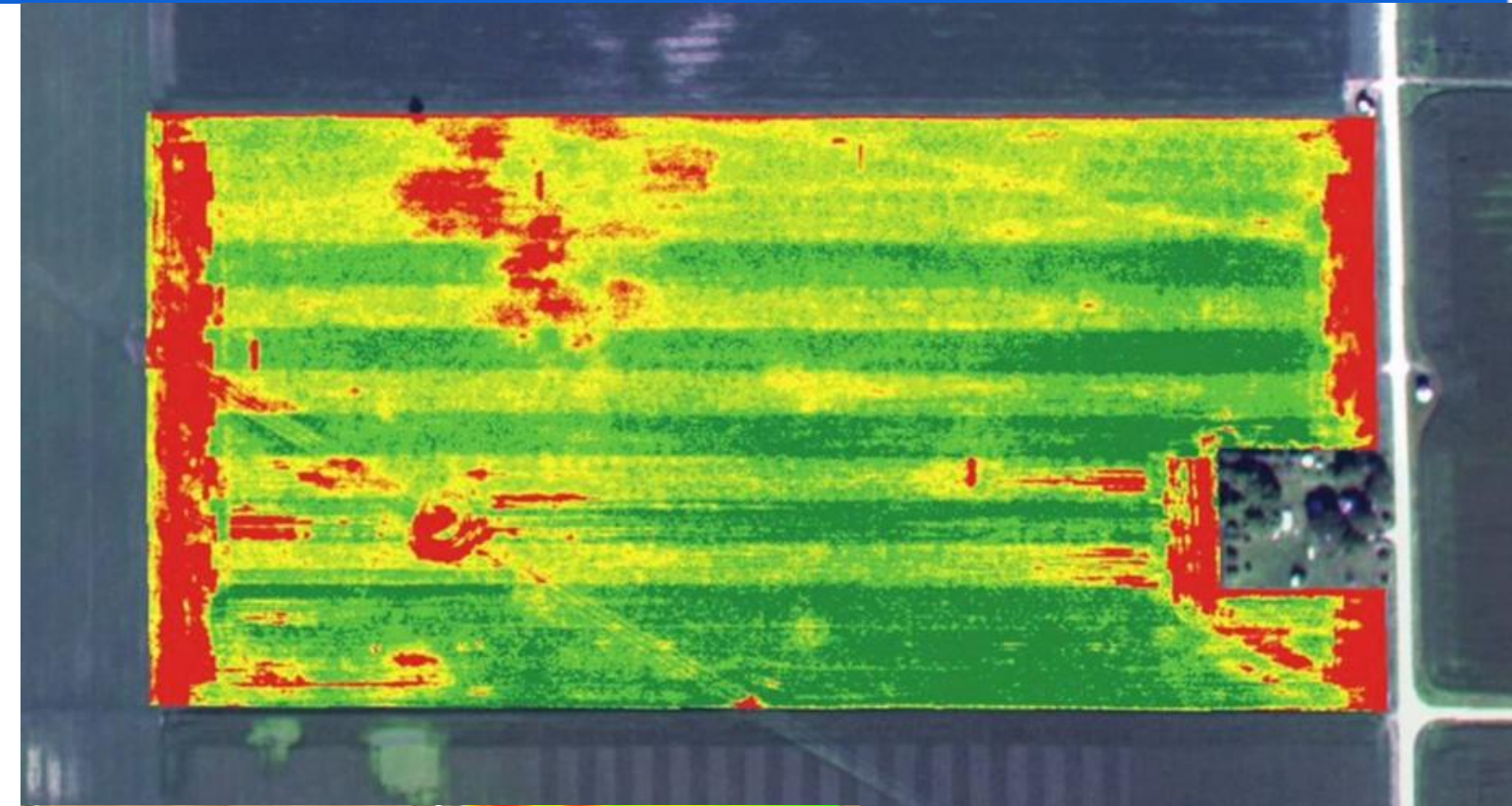
In this lesson, our drone's mission is to find a specific problem area in the field and take a photo of it.

- The drone will fly a pre-defined pattern to scan the field.
- The QR code placed in the floor map area with unhealthy crops.
- Once the QR is detected, the drone will stop and take a picture for later analysis and return to land
- This process simulates how farmers can identify and act on issues quickly.



Drone Flight Path

- To scan the field, the drone follows a systematic flight path called a “lawnmower” or grid pattern.
- This ensures full area coverage without missing spots. The drone flies in straight lines, back and forth across the field. Altitude, speed, and spacing between lines are pre-set in the flight plan.
- This pattern helps the drone gather consistent, accurate data across the whole field.

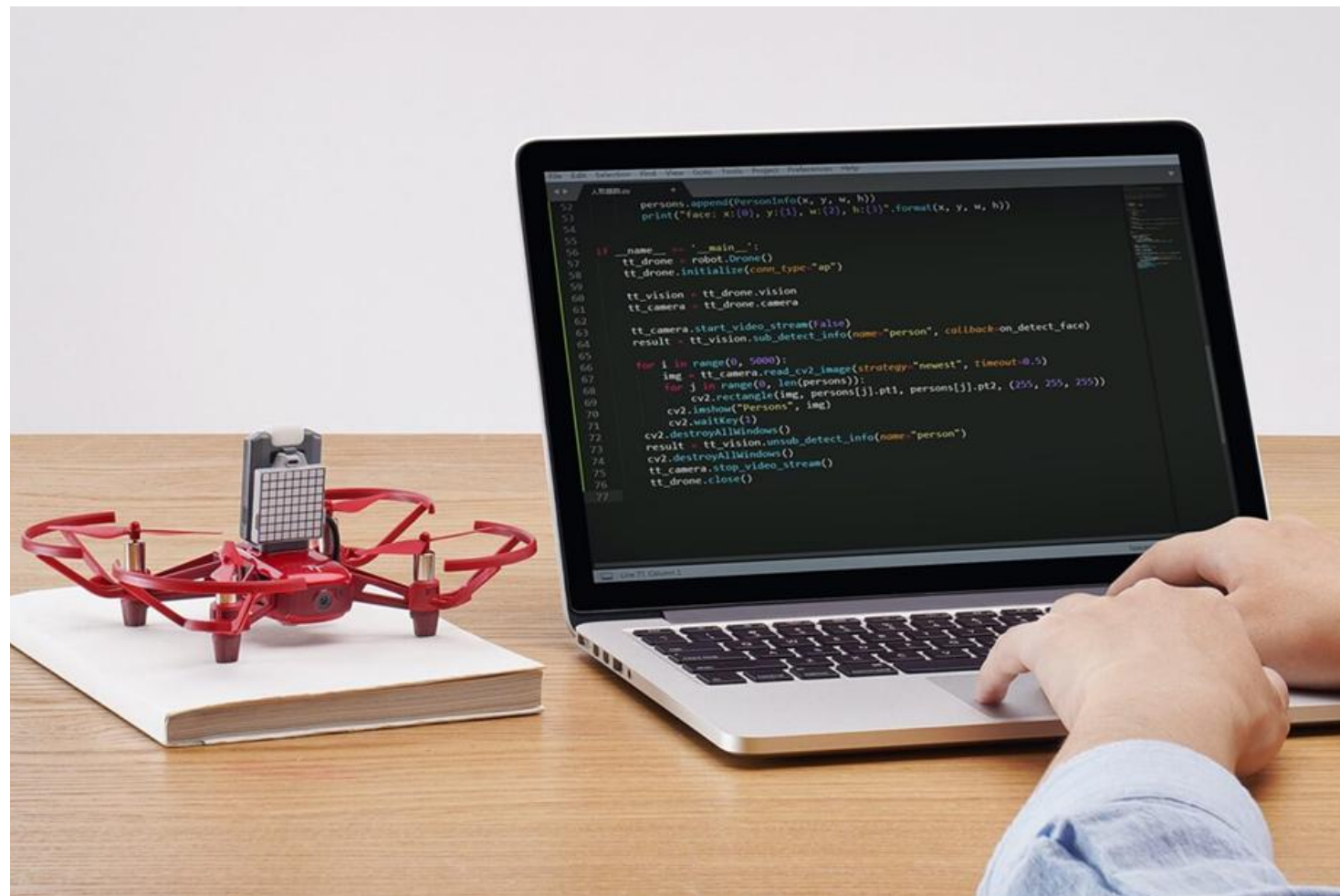


Activity : Programming Image Capture

- Take off.
- Perform a search pattern over a predefined area
- Find bad crops
- Take picture and land
- Return to home and land.



Materials Needed



HARDWARE

DJI Robomaster TT Drone

Drones4STEM FloorMat

PC SOFTWARE

MIND+ Scratch GUI Platform (PYTHON MODULE)

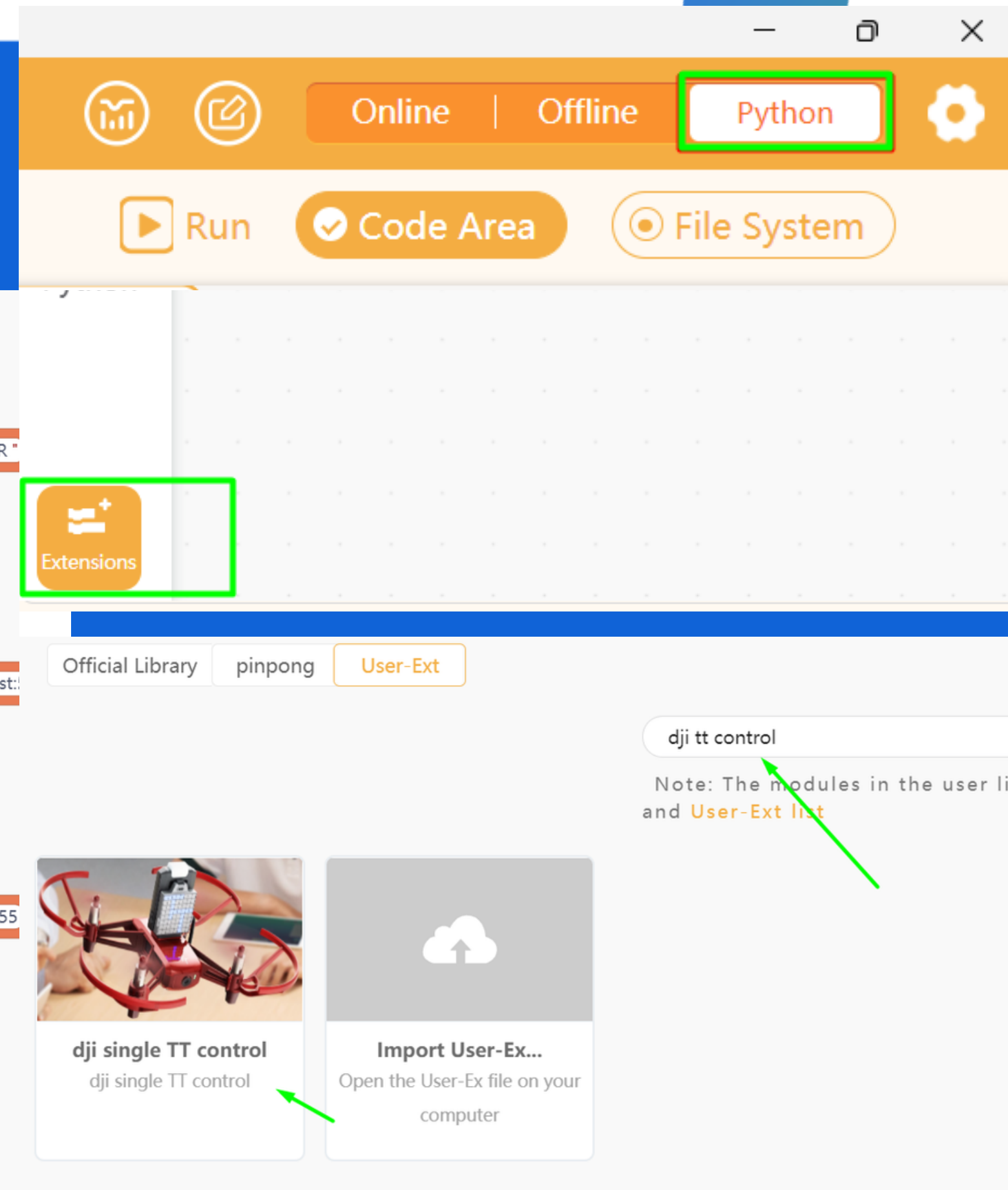
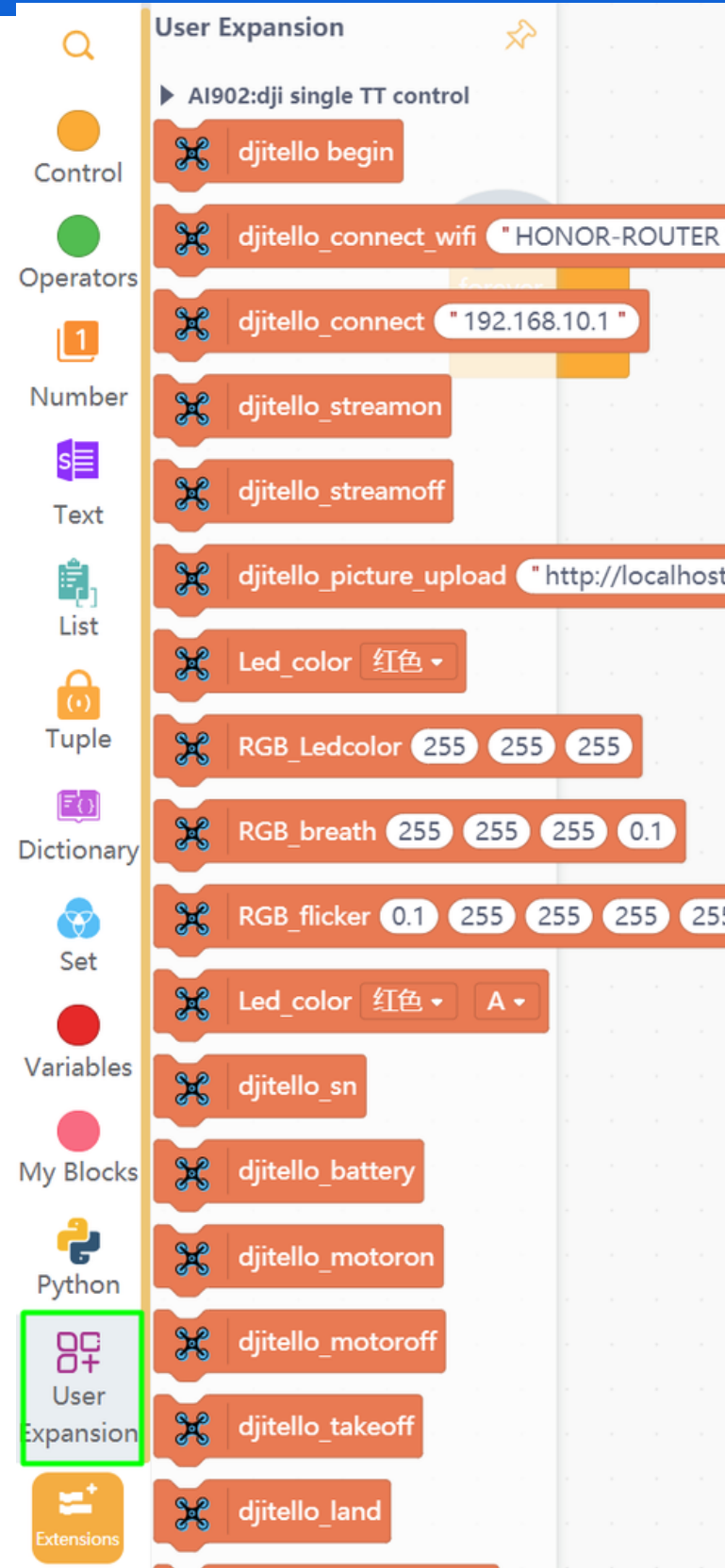
or

Drone4Stem Application

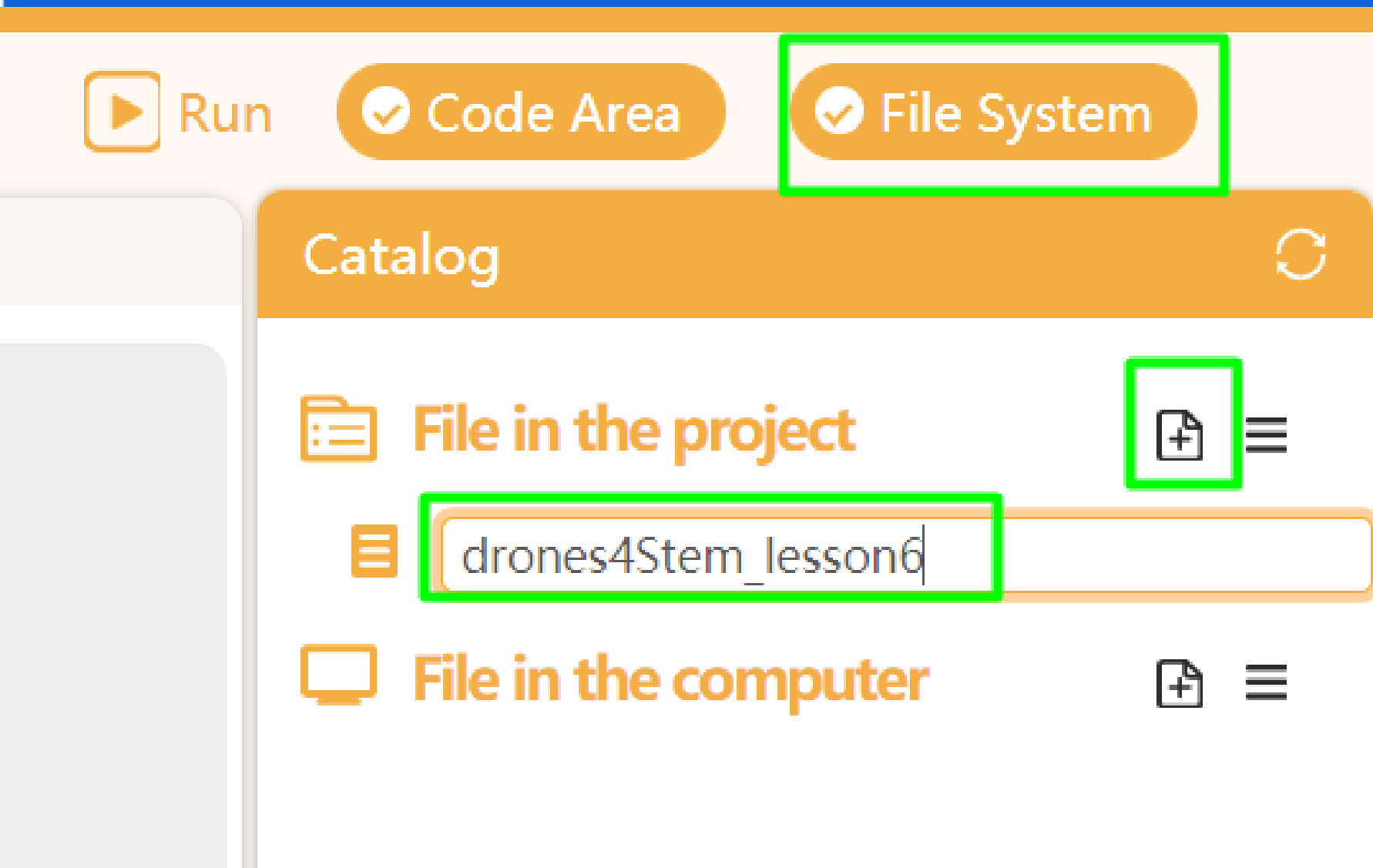
MIND+ Python Install

Installation Checklist Python

- Choose Python Module
- Choose EXTENSIONS->USER-EXT and search : DJI TT CONTROL and select the library
- USER EXPANSION block
 - TELLO TT Specific functions



Start a python program



- Click File System
- Add file (File in the project)
- Type a name for the file

Blocks to Python Code Generation

Mind+ Converts Blocks to Python code

We can copy and improve that in CODE Mode.

The screenshot displays the Mind+ IDE interface. On the left, a sidebar contains various tool categories: Control, Operators, Number, Text, List, Tuple, and Dictionary. The main workspace is divided into two tabs: 'Blocks' and 'Code'. The 'Blocks' tab is active, showing a script for a Tello drone. The script starts with a 'Python main program start' block, followed by 'djitello begin', 'djitello_connect' (with IP '192.168.10.1'), 'djitello_streamon', and 'djitello_takeoff'. These four blocks are enclosed in a 'forever' loop, which also contains 'djitello_challenge' and 'djitello_moveforward' (with distance 20). A green box highlights the first four blocks, and a green arrow points from the 'djitello_streamon' block to the 'Auto Generate' button. The 'Auto Generate' button is located in the top right corner of the workspace. To the right of the 'Auto Generate' button, the 'Code' tab is active, showing the generated Python code. The code is as follows:

```
1  # -*- coding: UTF-8 -*-
2
3  # MindPlus
4  # Python
5  import requests
6  from PIL import Image
7  import io
8  from djitellopy import Tello
9  import cv2
10 from djitellopy import TelloSwarm
11
12
13 tello = Tello(host="192.168.10.1")
14 tello.connect()
15 tello.streamon()
16 tello.takeoff()
17
18 while True:
19     tello.enable_mission_pads()
20     tello.set_mission_pad_detection_direction(0)
21     tello.move_forward(20)
22
```

On the far right, a 'Catalog' panel shows the project structure, including 'File in the project' and 'File in the computer' sections. The 'File in the project' section lists 'drones4stem_lesson6.py'. The 'Drones 4STEM' logo is visible in the bottom right corner.

Blocks to Python Code Generation

The screenshot displays the Mind+ V1.8.0 RC3.1 interface. The top bar includes the Mind+ logo, 'Project' and 'Learning' dropdowns, and buttons for 'Online', 'Offline', and 'Python'. Below this, a 'Blocks' tab is highlighted with a green box, and a 'Code' tab is also visible. The main workspace shows a Python script for a Tello drone mission. The script includes imports for requests, PIL Image, io, djitellopy Tello, cv2, and djitellopy TelloSwarm. It initializes a Tello object, connects it, streams video, and takes off. A while loop enables mission pads, sets the detection direction, and moves forward 20 units. On the right, a 'Catalog' panel shows 'File in the project' and 'File in the computer' sections. The 'File in the project' section has a green box around 'drones4Stem_lesson6.py'. The bottom panel is a 'Terminal' window showing the Python 3.8.5 prompt.

```
1  # -*- coding: UTF-8 -*-
2
3  # MindPlus
4  # Python
5  import requests
6  from PIL import Image
7  import io
8  from djitellopy import Tello
9  import cv2
10 from djitellopy import TelloSwarm
11
12
13 tello = Tello(host="192.168.10.1")
14 tello.connect()
15 tello.streamon()
16 tello.takeoff()
17
18 while True:
19     tello.enable_mission_pads()
20     tello.set_mission_pad_detection_direction(0)
21     tello.move_forward(20)
22
```

Terminal Clear output

```
Python 3.8.5 [MSC v.1924 64 bit AMD64] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> _
```

Drones4Stem Floor Mat Specs



- Every Grid is 40x40cm
- Forward Movement : 100 cm
- Change Lane: 40 cm slip
- QR Code 2 is bad irrigation
- QR Code 4 is bad crops

Initialization and Taking Off

Initialize

```
tello = Tello()
tello.connect()
print(f"Battery: {tello.get_battery()}%")
```

Start video stream

```
tello.streamon()
frame_read = tello.get_frame_read()
```

Enable Mission Pads

```
tello.enable_mission_pads()
tello.set_mission_pad_detection_direction(0) # Downward detection
```

Take off

```
tello.takeoff()
time.sleep(2)
```

Check for mission pads function, photo, return home

```
# Define helper to check for Pad 4
def check_mission_pad():
    pad = tello.get_mission_pad_id()
    print(f"Detected Pad: {pad}")

    if pad == 4:
        print("Pad 4 detected - taking photo and returning to Pad 1")

        # Take photo
        frame = frame_read.frame
        cv2.imwrite("pad4_photo.jpg", frame)
        print("Photo saved as pad4_photo.jpg")

        # Move to return path
        tello.move_right(80)
        tello.move_back(80)
```

```
# Try to find Pad 1
print("Looking for Pad 1 to land...")
for _ in range(10): # Retry up to 10 times
    pad1 = tello.get_mission_pad_id()
    print(f"Pad ID: {pad1}")
    if pad1 == 1:
        print("Pad 1 found - landing.")
        tello.land()
        tello.streamoff()
        exit()
        time.sleep(1)

print("Pad 1 not found. Landing anyway.")
tello.land()
tello.streamoff()
exit()
```

Lawnmower Pattern

```
# Flight pattern with Pad 4 check
tello.move_forward(120)
check_mission_pad()

tello.move_left(40)
check_mission_pad()

tello.move_back(120)
check_mission_pad()

tello.move_left(40)
check_mission_pad()

tello.move_forward(120)
check_mission_pad()

# End of pattern
print("Pattern complete. Mission Pad 4 not found. Landing.")
tello.land()
tello.streamoff()
```

Indoor Drone Flying Safety Rules

1 Before the Flight:

- Inspect the Drone
 - Check for damage or loose propellers.
 - Ensure the battery is fully charged.
- Prepare the Area
 - Remove obstacles like chairs, desks, and other items.
 - Ensure a spacious and open environment.
 - Close Windows or air stream sources
- Inform Participants
 - Brief everyone on safety protocols.
 - Maintain a safe distance of at least 2 meters from the drone.

Indoor Drone Flying Safety Rules

2 During the Flight:

- Controlled Takeoff and Landing
 - Launch and land on a flat, stable surface.
 - Avoid flying near people or objects.
- Flight Boundaries
 - Keep the drone within visual line-of-sight.
 - Fly at low altitudes to avoid ceiling impact.
 - Land when battery <20%
- Emergency Protocol
 - Stop the drone immediately if it veers out of control. (SPACE initiates LANDING)
 - Use the "emergency stop" button on the app if needed. ("e" stops motors)

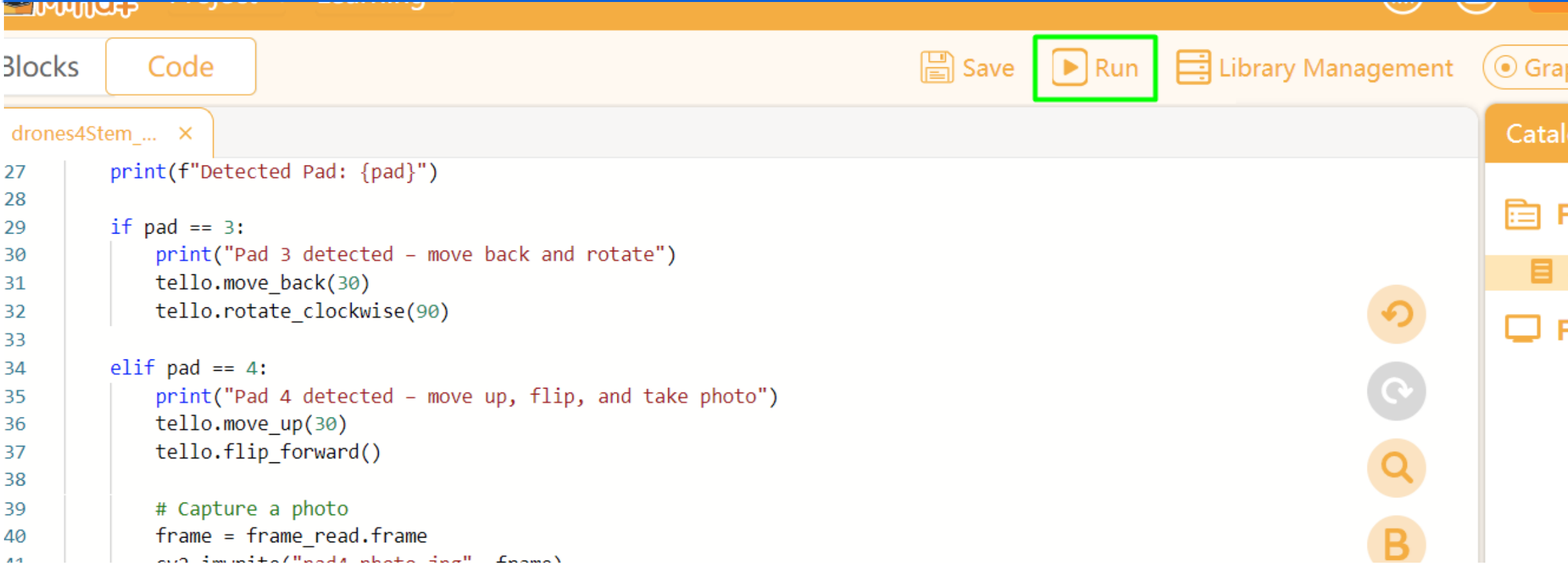
Indoor Drone Flying Safety Rules

3 After the Flight:

- Power Off Safely
 - Turn off the drone and disconnect the battery.
- Inspect for Damage
 - Check for any issues after flight.
- Store Safely
 - Keep the drone in a secure, dry place.

ALWAYS STAY CALM AND FOCUSED

Test the Program



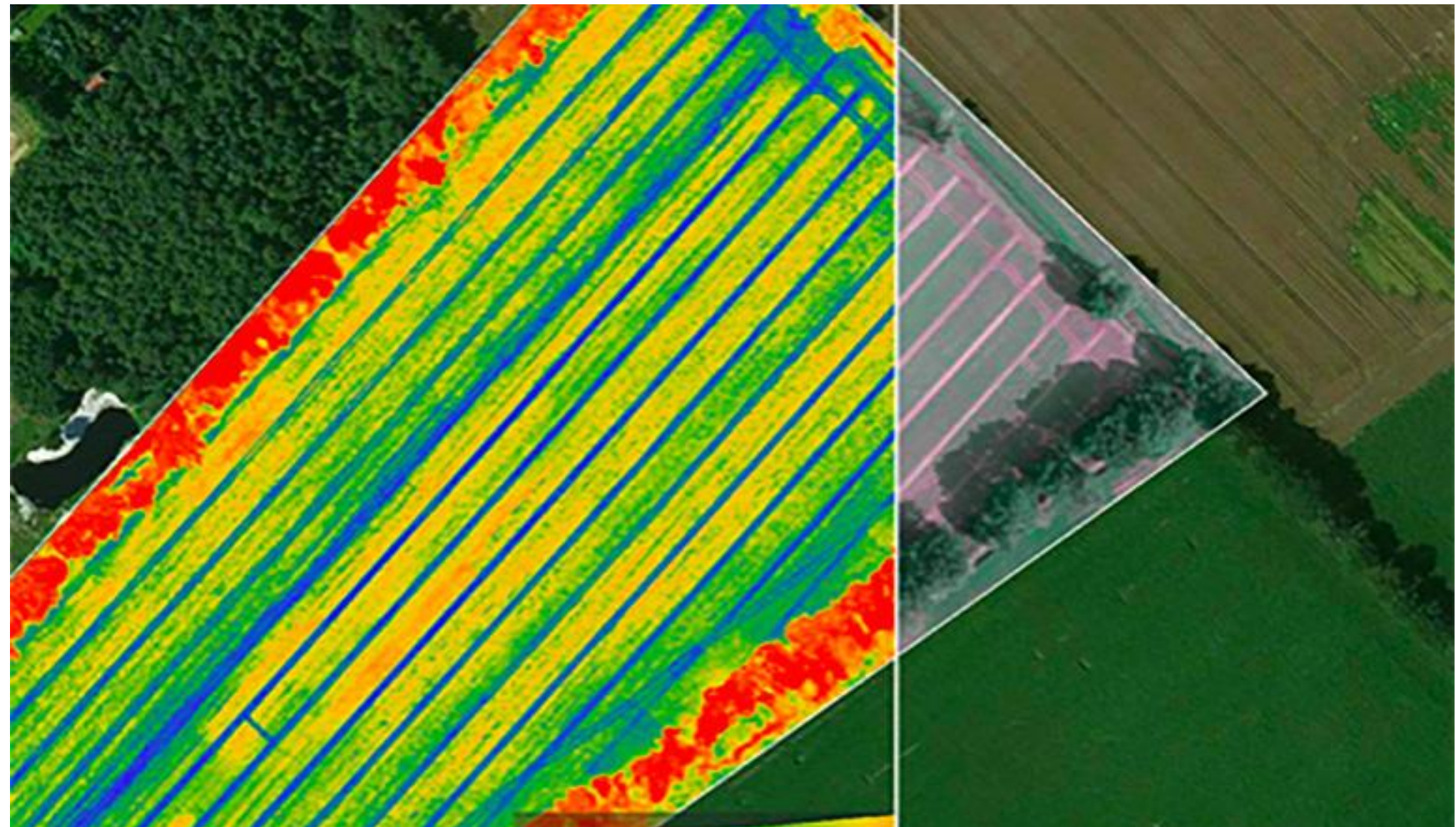
The screenshot shows the Drone4STEM web interface. At the top, there are tabs for 'Blocks' and 'Code'. The 'Code' tab is active, displaying a Python script for controlling a drone. The script includes comments and function calls for moving the drone, rotating it, and capturing a photo. The 'Run' button, represented by a play icon, is highlighted with a green box. Other buttons like 'Save', 'Library Management', and 'Graph' are also visible. On the right side, there is a sidebar with a 'Catalog' section and a list of folders. At the bottom right, there is a vertical toolbar with icons for undo, redo, search, and a button labeled 'B'.

```
27 print(f"Detected Pad: {pad}")
28
29 if pad == 3:
30     print("Pad 3 detected - move back and rotate")
31     tello.move_back(30)
32     tello.rotate_clockwise(90)
33
34 elif pad == 4:
35     print("Pad 4 detected - move up, flip, and take photo")
36     tello.move_up(30)
37     tello.flip_forward()
38
39     # Capture a photo
40     frame = frame_read.frame
41     img_write("pad4_photo.jpg", frame)
```



Basic Image Analysis

- Healthy crops: Bright green colors.
- Problem areas: Yellow or brown spots, wilting leaves.
- Soil issues: Dry or cracked ground.



Improvements and Limitations

- What will happen if the Mission Pad isn't recognized?
- Can you add some led pattern (i.e red for bad crops)
- What happens if battery is going low?
- Why the drone increase altitude before taking photo?



RECAP

- What We Learned:
 - Why drone is important in smart farming.
 - How to install python module.
 - What are the basics command to initiliazze DJI Tello with python.
 - How to make a function to read mission pads.
 - How to make a lawnmover path.
- Safety First:
 - Importance of clearing the flying area.
 - Maintaining a safe environment for everyone.



Additional Resources

Further Reading

- <https://djitellopy.readthedocs.io/en/latest/tello/>

Videos

[How to install python module](#)



Thank you for your
attention !



Co-funded by
the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.