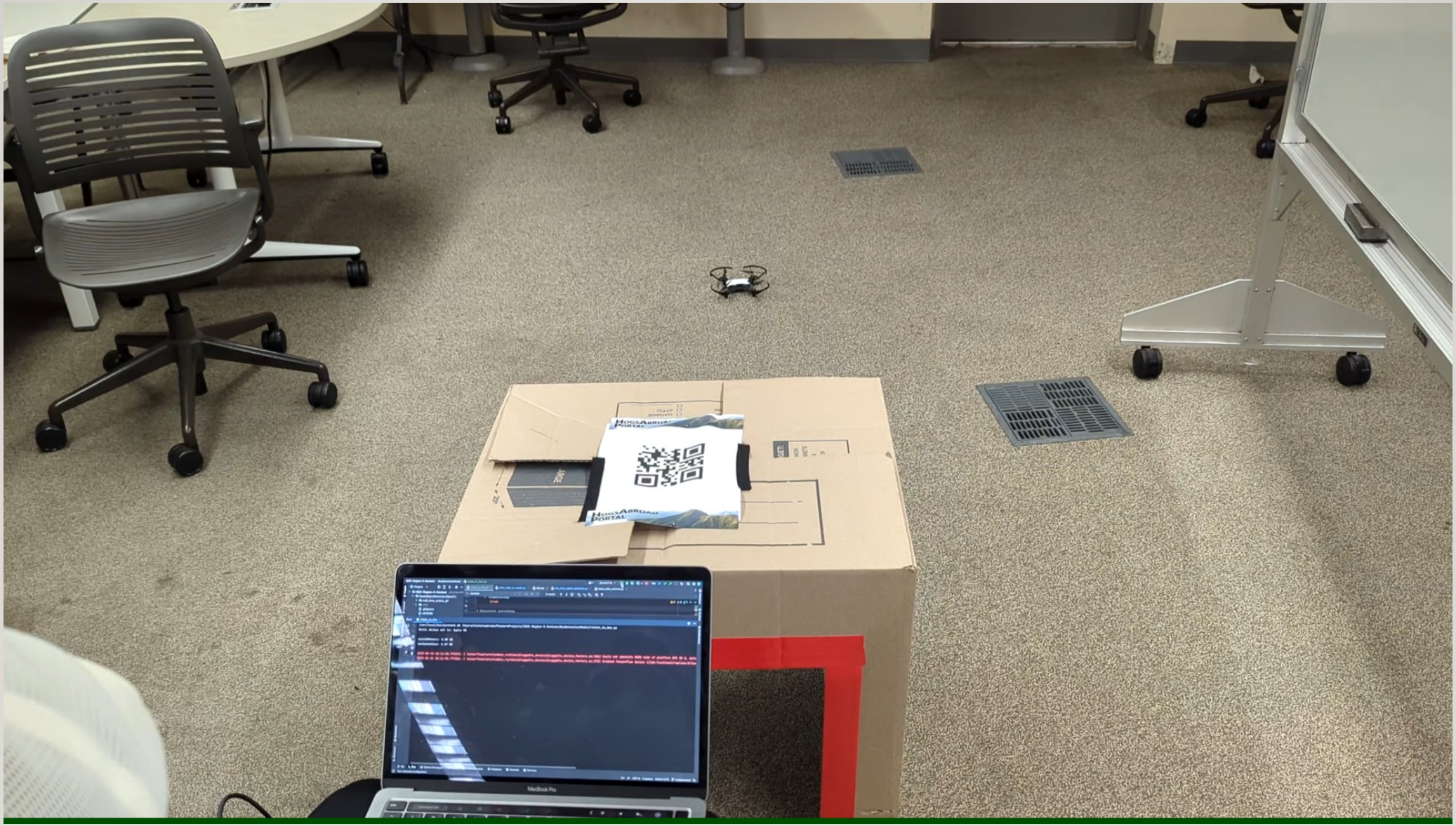
TEAM 8 EE IEEE ROBOTICS COMPETITION

NICHOLAS BROWN, CALLUM BRUTON, JASE CORNETT, AUSTIN FLYNN, STEPHANIE STOCK

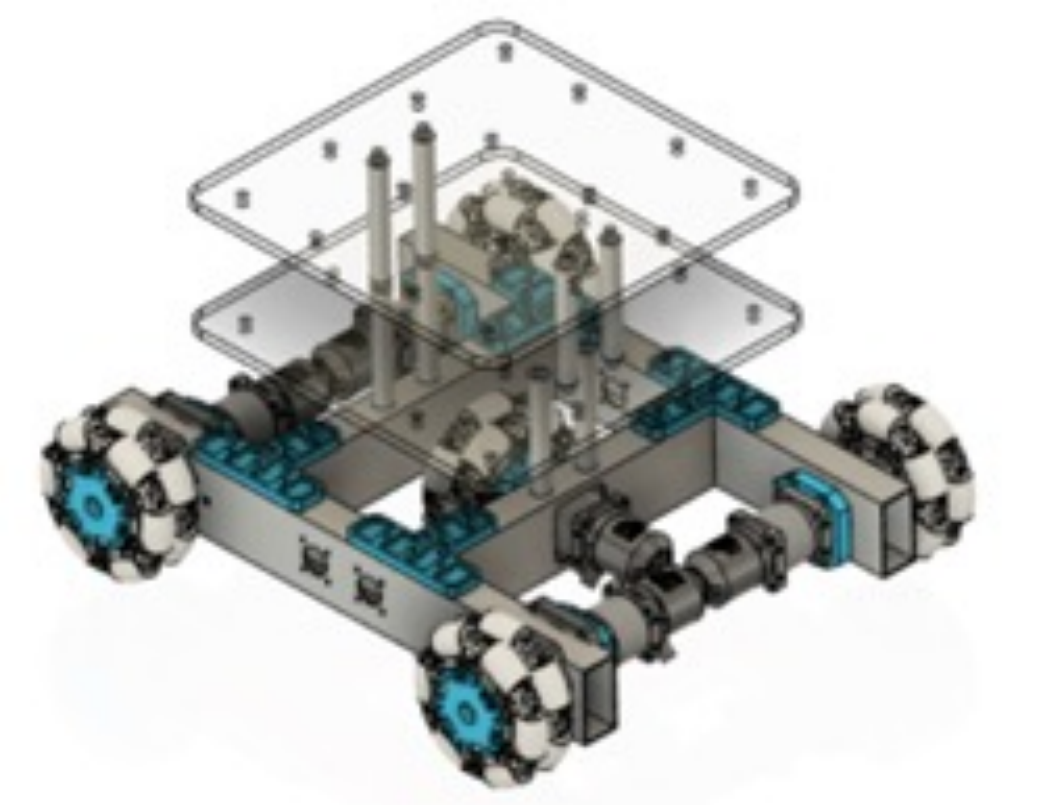
DJI RYZE TELLO DRONE

- Has a pre-made Python SDK with function that allow us to control the drone
- Uses a finite-state machine to operate autonomously
- Reads QR codes using OpenCV with a pre-made QR reader
- Uses object detection model to locate box or ground robot and fly to it
- Sends crucial data back to Nvidia Jetson over UDP



GROUND ROBOT

- Aluminum and Acrylic Frame
- Five 4" Omni-wheels
- Sensors
  - Ultrasonic
  - Intel RealSense D415i
  - Raspberry Pi Camera V2



IEEE-Region-5-Contest 〉 ascii_realsense.py

Current File | Git:

Project | README.md | basic_realsense.py | depth_realsense.py | ascii_realsense.py

Run: ascii_realsense

```
C:\Users\steph\AppData\Local\Programs\Python\Python310\python.exe C:\Users\steph\PycharmProjects\IEEE-Region-5-Contest\ascii_realsense.py
```

Project | Commit | Pull Requests | Bookmarks | Structure | Notifications

Git | Run | Python Packages | TODO | Python Console | Problems | Terminal | Services

Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built Python packages shared indexes // Always download // Download once... (today 10:49 | 26:21 | CRLF | UTF-8 | 4 spaces | Python 3.10 (IEEE-Region-5-Contest) | stephanie/realsense

NVIDIA JETSON NANO

- Main processing unit for done and ground robot.
- Takes input from the drone using UDP, Raspberry Pi Camera module V2, and ultrasonic sensor
- Outputs commands to the drone and ground robot using Python
- I/O used for this project include wireless networking, USB, CSI connector, and GPIO pins

sketch_mar9b | Arduino IDE 2.0.3

File Edit Sketch Tools Help

Arduino MKR1000

Serial Monitor

sketch_mar9b.ino

```
// defines pins numbers
const int trigPin = 7;
const int echoPin = 6;
// defines variables
long duration;
int distance;
void setup() {
  pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
  pinMode(echoPin, INPUT); // Sets the echoPin as an Input
  Serial.begin(9600); // Starts the serial communication
}
void loop() {
  // Clears the trigPin
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  // Sets the trigPin on HIGH state for 10 micro seconds
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  // Reads the echoPin, returns the sound wave travel time in microseconds
  duration = pulseIn(echoPin, HIGH);
  // Calculating the distance
  distance = duration * 0.034 / 2;
  // Prints the distance on the Serial Monitor
  Serial.print("Distance: ");
  Serial.println(distance);
  delay(250);
}
```

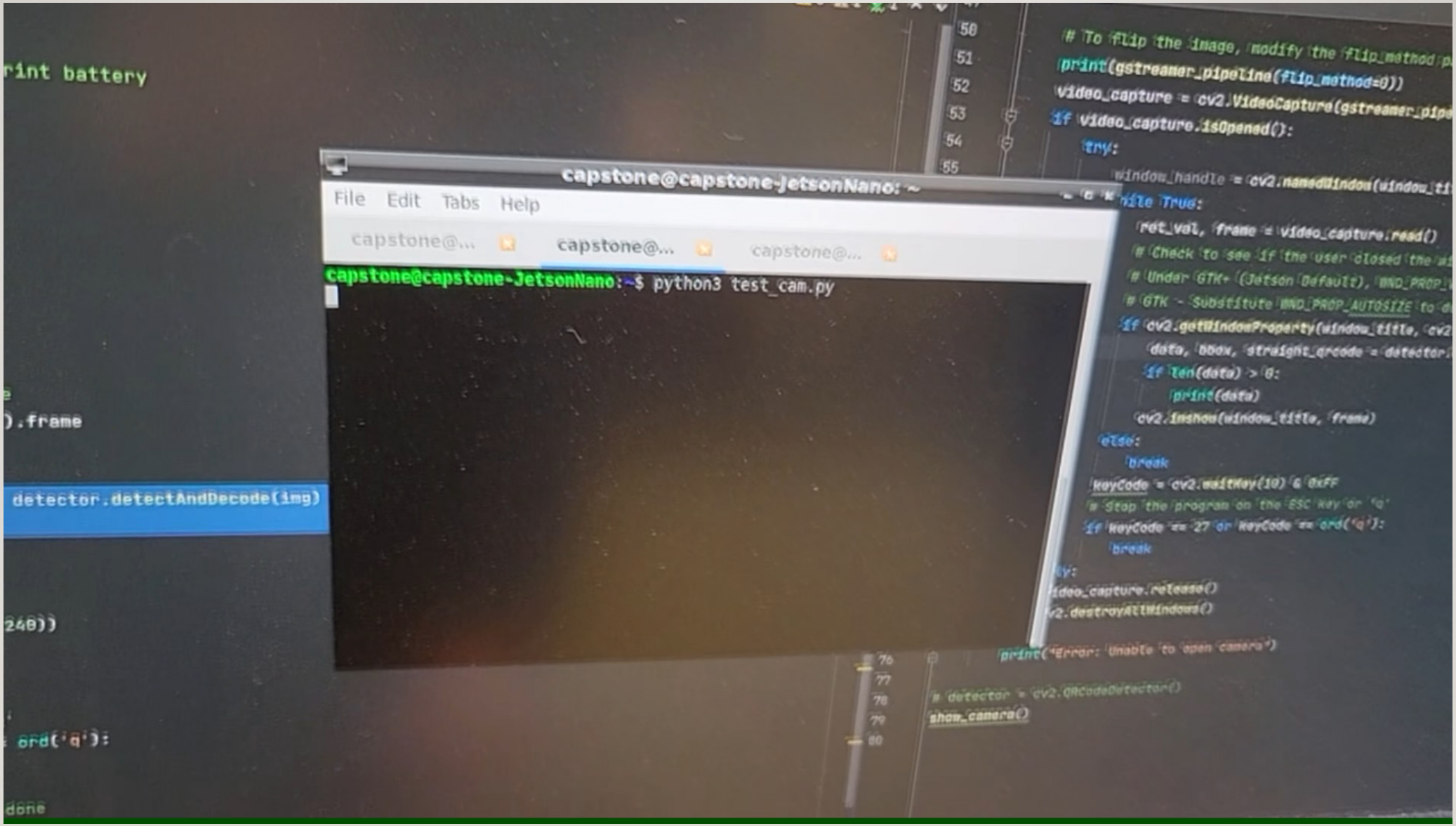
Output

```
[==========================   ] 91% (192/209 pages)
[==============================] 100% (209/209 pages)
done in 0.104 seconds

Verify 13372 bytes of flash with checksum.
Verify successful
done in 0.017 seconds
CPU reset.
```

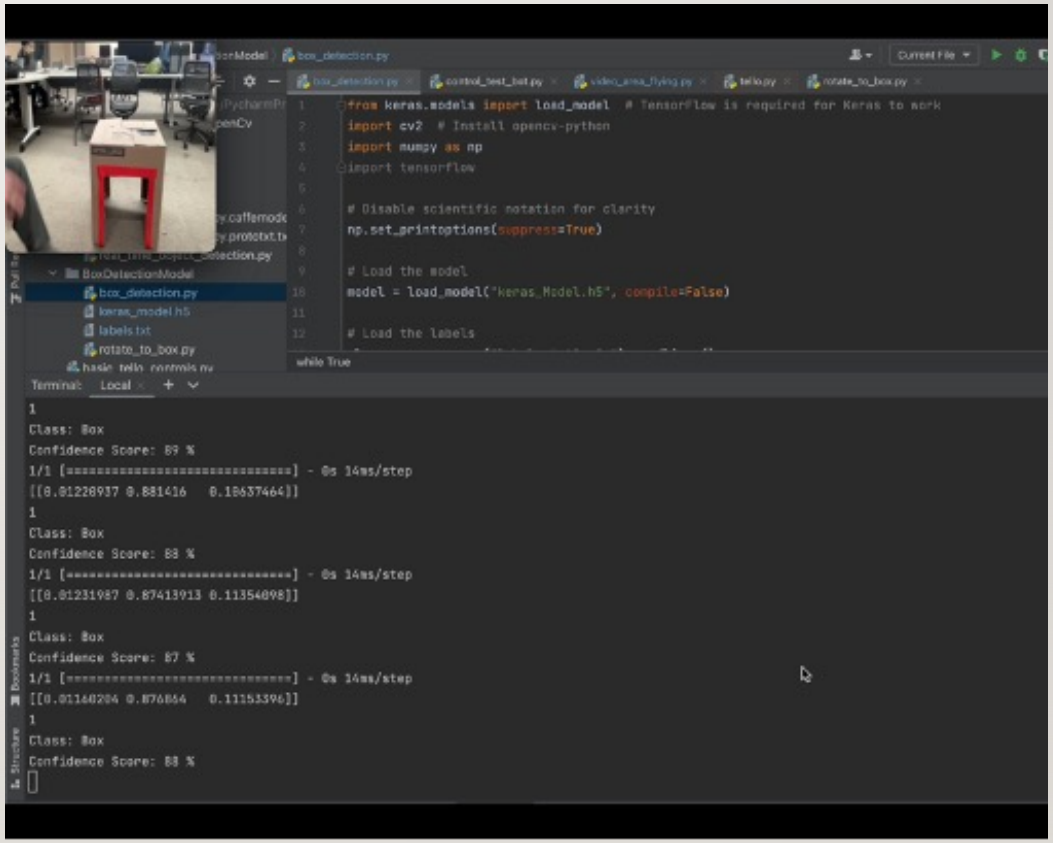
Ln 27, Col 12 UTF-8 Arduino MKR1000 on COM3

4:27 PM 3/9/2023


capstone@capstone-JetsonNano: ~
File Edit Tabs Help
capstone@...
capstone@...
capstone@...
capstone@capstone-JetsonNano:~$ python3 test_cam.py
detector.detectAndDecode(img)
print(gstreamer_pipeline(flip_method=0))
video_capture = cv2.VideoCapture(gstreamer_pipe
if video_capture.isOpened():
try:
while True:
ret_val, frame = video_capture.read()
if len(data) > 0:
print(data)
cv2.imshow(window_title, frame)
else:
break
keyCode = cv2.waitKey(10) & 0xFF
if keyCode == 27 or keyCode == ord('q'):
break
video_capture.release()
cv2.destroyAllWindows()
print("Error: Unable to open camera")
# detector = cv2.QRCodeDetector()
show_camera()

OBJECT DETECTION MODEL

- Created using Teachable Machines
- Exported for Keras that can be used in Python with supporting packages
- Can differentiate between the boxes, drones, and everything else
- Is used with the drone's camera and on the Intel RealSense



IEEE-Region-5-Contest 〉 BoxDetectionModel 〉 box_detection.py

box_detection.py | control_test_bot.py | video_area_flying.py | tello.py | rotate_to_box.py

```python
from keras.models import load_model  # TensorFlow is required for Keras to work
import cv2  # Install opencv-python
import numpy as np
import tensorflow

# Disable scientific notation for clarity
np.set_printoptions(suppress=True)

# Load the model
model = load_model("keras_Model.h5", compile=False)

# Load the labels
```

```
nicholasbrown@nicholass-mbp BoxDetectionModel % python3.10 box_detection.py
Metal device set to: Apple M2

systemMemory: 8.00 GB
maxCacheSize: 2.67 GB

2023-03-09 16:21:18.070658: I tensorflow/core/common_runtime/pluggable_device/pluggable_device_factory.cc:306] Could not identify NUMA node of platform GPU ID 0, defaulting
 to 0. Your kernel may not have been built with NUMA support.
2023-03-09 16:21:18.071055: I tensorflow/core/common_runtime/pluggable_device/pluggable_device_factory.cc:272] Created TensorFlow device (/job:localhost/replica:0/task:0/de
vice:GPU:0 with 0 MB memory) -> physical PluggableDevice (device: 0, name: METAL, pci bus id: <undefined>)
```