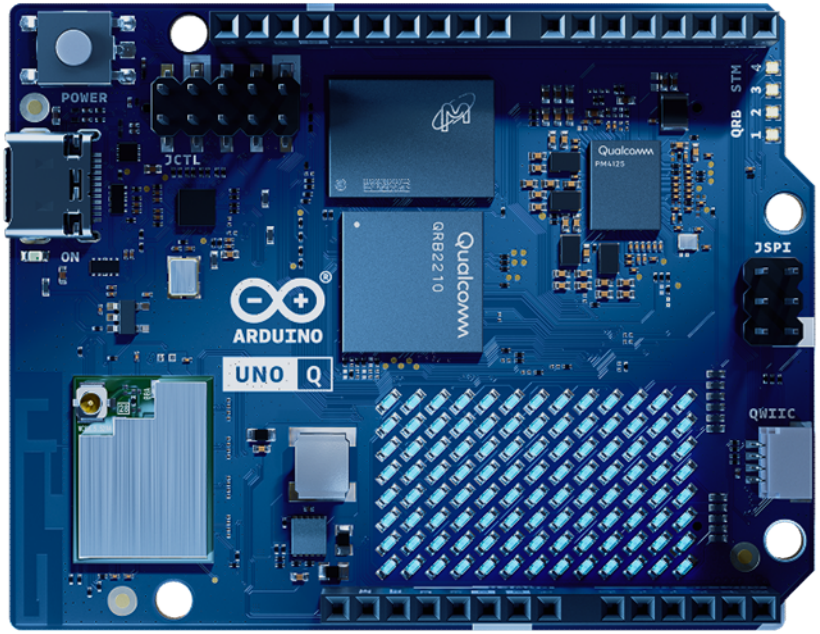
Bricks

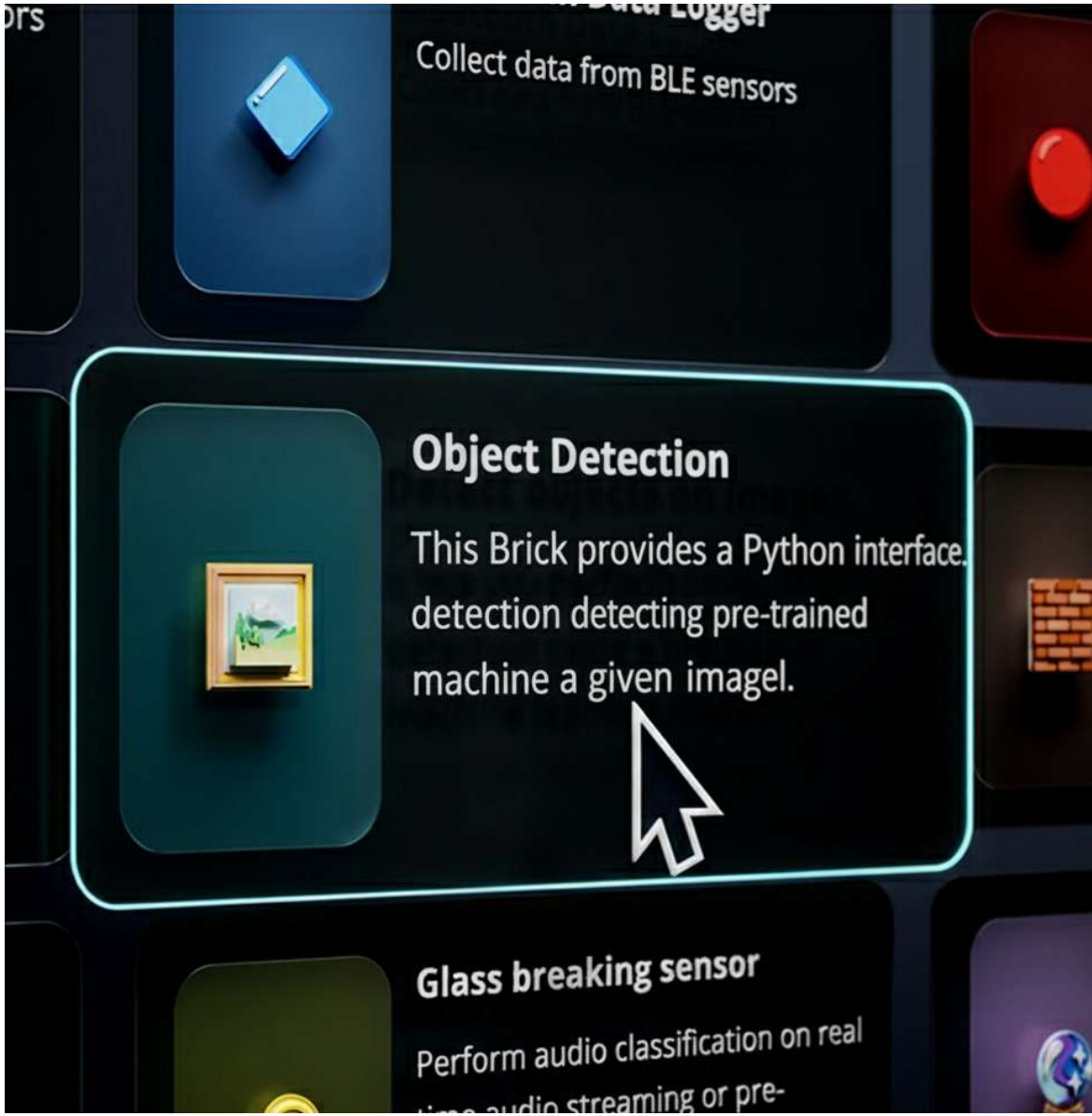
ARDUINO

Object Detection

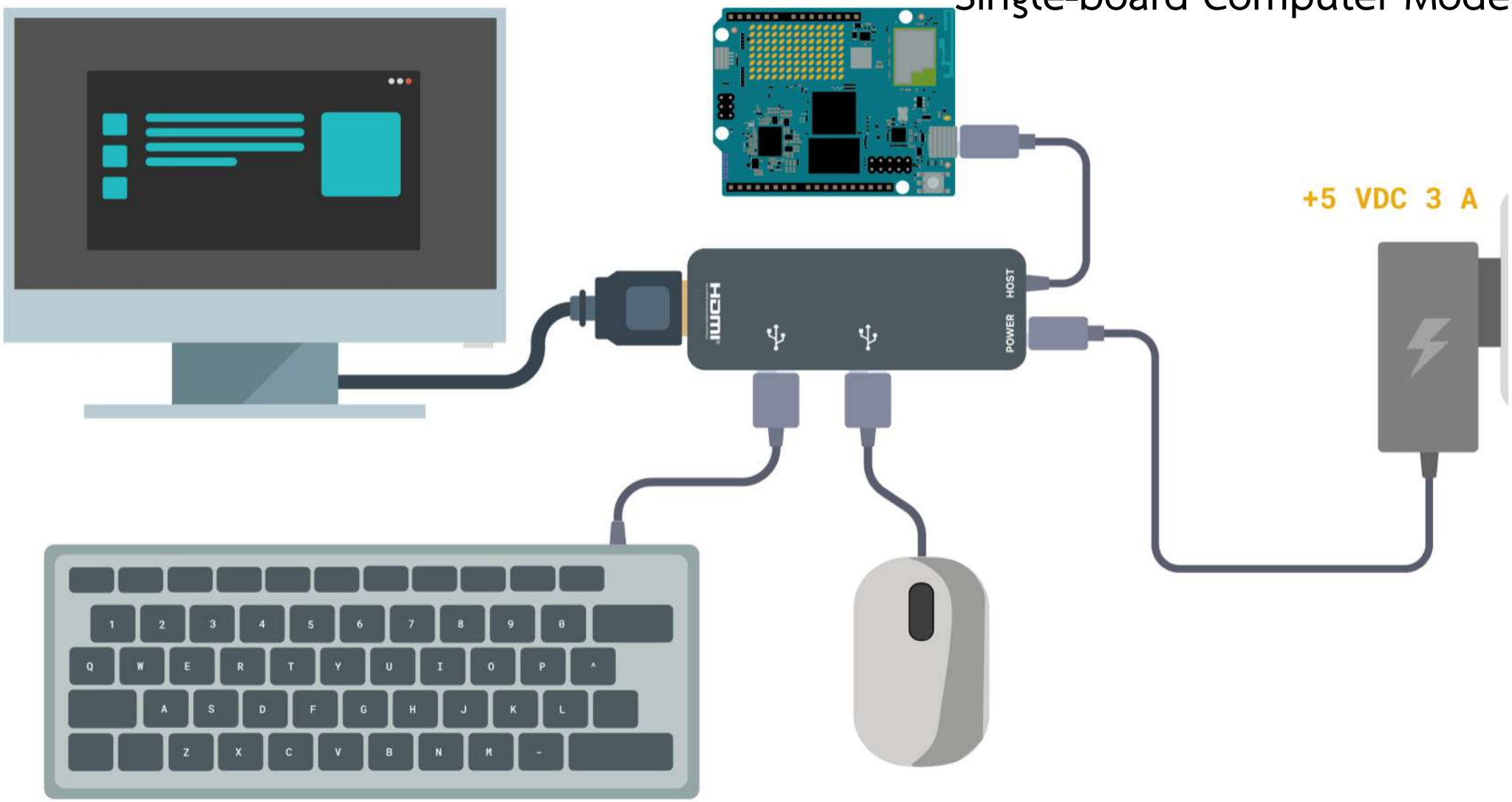
UNO Q

ตรวจจับวัตถุ

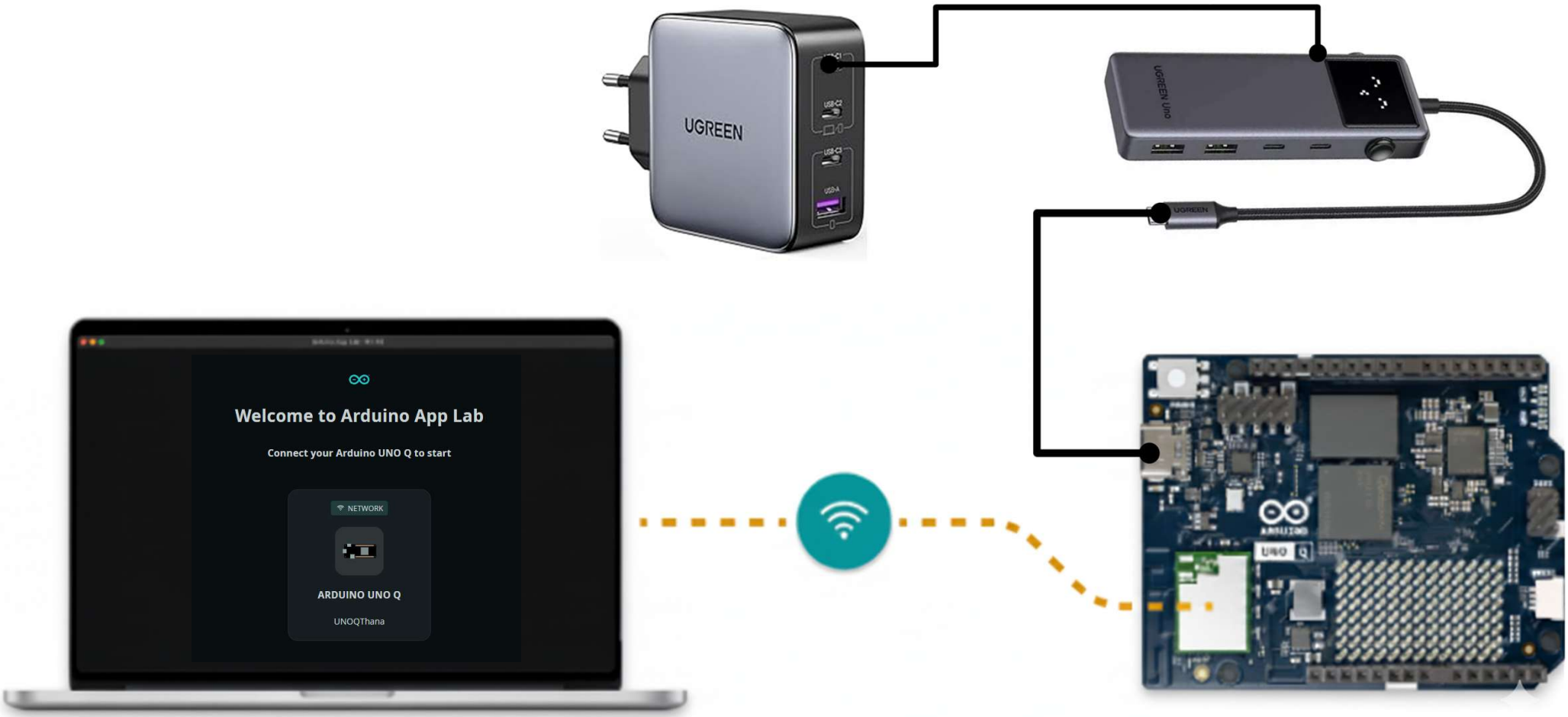




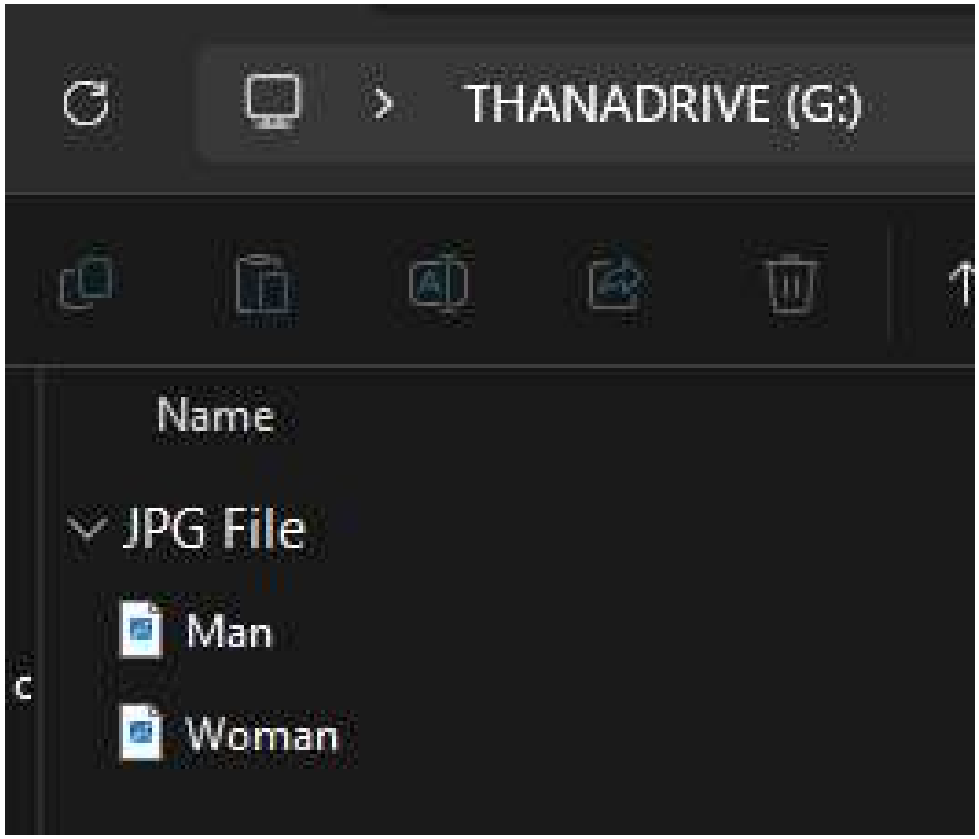
Single-board Computer Mode

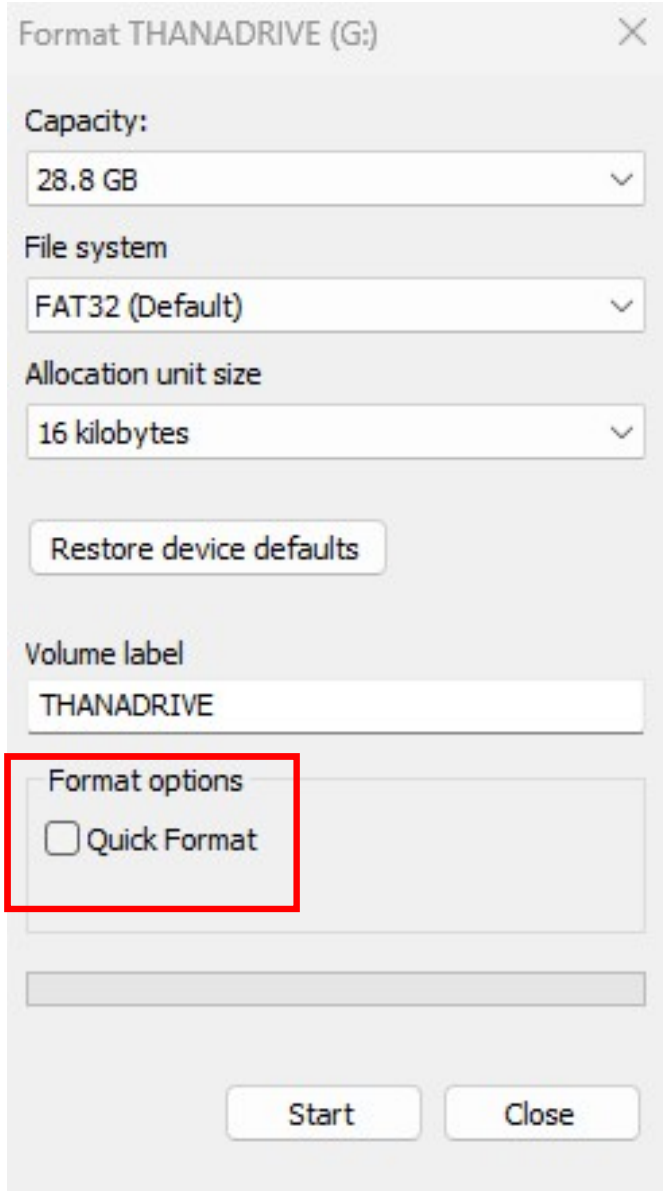


Network Mode



การฟอร์แม็ตไดร์ฟ





ตรวจสอบแฟลชไดร์ฟ

```
arduino@ThanaUNOQ:~$ lsblk
NAME          MAJ:MIN RM   SIZE RO TYPE MOUNTPOINTS
sda             8:0    1  28.9G  0 disk
└─sda1          8:1    1  28.9G  0 part
mmcblk0       179:0    0  14.6G  0 disk
├─mmcblk0p1   179:1    0   3.5M  0 part
├─mmcblk0p2   179:2    0   3.5M  0 part
arduino@ThanaUNOQ:~$ lsblk
NAME          MAJ:MIN RM   SIZE RO TYPE MOUNTPOINTS
sdb             8:16   1  14.3G  0 disk
└─sdb1          8:17   1  14.3G  0 part
```

ตรวจสอบ File System ของแฟล็ชไดร์ฟ

```
arduino@ThanaUNOQ:~$ sudo blkid /dev/sda1
/dev/sda1: LABEL_FATBOOT="THANADRIVE" LABEL="THANADRIVE" UUID="F6D7-13F1" BLOCK_SIZE="512" TYPE="vfat" PARTUUID="c260cf0
b-01"
```

สร้างโฟลเดอร์สำหรับ mount

```
arduino@ThanaUNOQ:~$ sudo mkdir -p /media/usb
[sudo] password for arduino:
```

สั่ง mount แฟลชไดรฟ์

```
arduino@ThanaUNOQ:~$ sudo mount /dev/sda1 /media/usb
```

ตรวจสอบว่า mount แล้วหรือยัง

```
arduino@ThanaUNOQ:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            845M     0  845M   0% /dev
tmpfs           175M  1.8M  173M   2% /run
/dev/mmcblk0p68 9.8G  5.0G  4.3G  54% /
tmpfs           871M   12K  871M   1% /dev/shm
efivarfs        128K  2.2K  126K   2% /sys/firmware/efi/efivars
tmpfs           5.0M     0  5.0M   0% /run/lock
tmpfs           871M  8.0K  871M   1% /tmp
tmpfs           1.0M     0  1.0M   0% /run/credentials/systemd-journald.service
/dev/mmcblk0p69 3.6G  673M  2.8G  20% /home/arduino
/dev/mmcblk0p67 488M  113M  375M  24% /boot/efi
tmpfs           175M   48K  175M   1% /run/user/103
tmpfs           1.0M     0  1.0M   0% /run/credentials/getty@tty1.service
tmpfs           1.0M     0  1.0M   0% /run/credentials/serial-getty@ttyMSM0.service
tmpfs           175M   44K  175M   1% /run/user/1000
/dev/sda1        15G  2.7M   15G   1% /media/usb
arduino@ThanaUNOQ:~$
```

คัดลอกไฟล์ไปยังโฟล์เดอร์ของ App ที่สร้าง

```
arduino@ThanaUNOQ:~$ ls
ArduinoApps  Desktop  Documents  Downloads  lost+found  Music  Pictures  Public  Templates  Videos
arduino@ThanaUNOQ:~$
```

/home/arduino/ArduinoApps/testing-object-detection-brick

```
arduino@ThanaUNOQ:~$ sudo cp /media/usb/Man.jpg /home/arduino/ArduinoApps/testing-object-detection-brick
```

```
arduino@ThanaUNOQ:~$ sudo cp /media/usb/*.jpg /home/arduino/ArduinoApps/testing-object-detection-brick
```

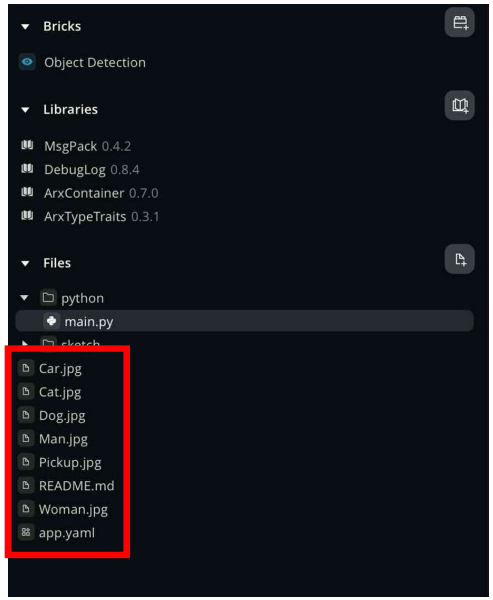
ตรวจสอบว่าไฟล์อยู่ในนั้นจริงหรือเปล่า

```
arduino@ThanaUNOQ:~$ sudo ls /home/arduino/ArduinoApps/testing-object-detection-brick
app.yaml  Car.jpg  Cat.jpg  Dog.jpg  Man.jpg  Pickup.jpg  python  README.md  sketch  Woman.jpg
arduino@ThanaUNOQ:~$
```

คัดลอกไฟล์ .png ไปยังโฟล์เดอร์ของ App ที่สร้าง พร้อมตรวจสอบว่าไฟล์อยู่ในนั้นจริงหรือเปล่า

```
arduino@ThanaUNOQ:~$ sudo cp /media/usb/*.png /home/arduino/ArduinoApps/testing-object-detection-brick
arduino@ThanaUNOQ:~$ sudo ls /home/arduino/ArduinoApps/testing-object-detection-brick
app.yaml  Car.jpg  Cat2.png  Cat.jpg  Dog2.png  Dog.jpg  Man.jpg  Pickup.jpg  python  README.md  sketch  Woman.jpg
```

/home/arduino/ArduinoApps/testing-object-detection-brick



ถ้าต้องการลบไฟล์ใช้คำสั่ง

```
rm ~/Pictures/Man.jpg
```

```
rm ~/Pictures/Man.jpg ~/Pictures/Dog.jpg ~/Pictures/Car.jpg
```

```
rm ~/Pictures/*.jpg
```

```
rm ~/Pictures/Man*.jpg
```

การนำแฟล็ชไดร์ฟออกจาก USB

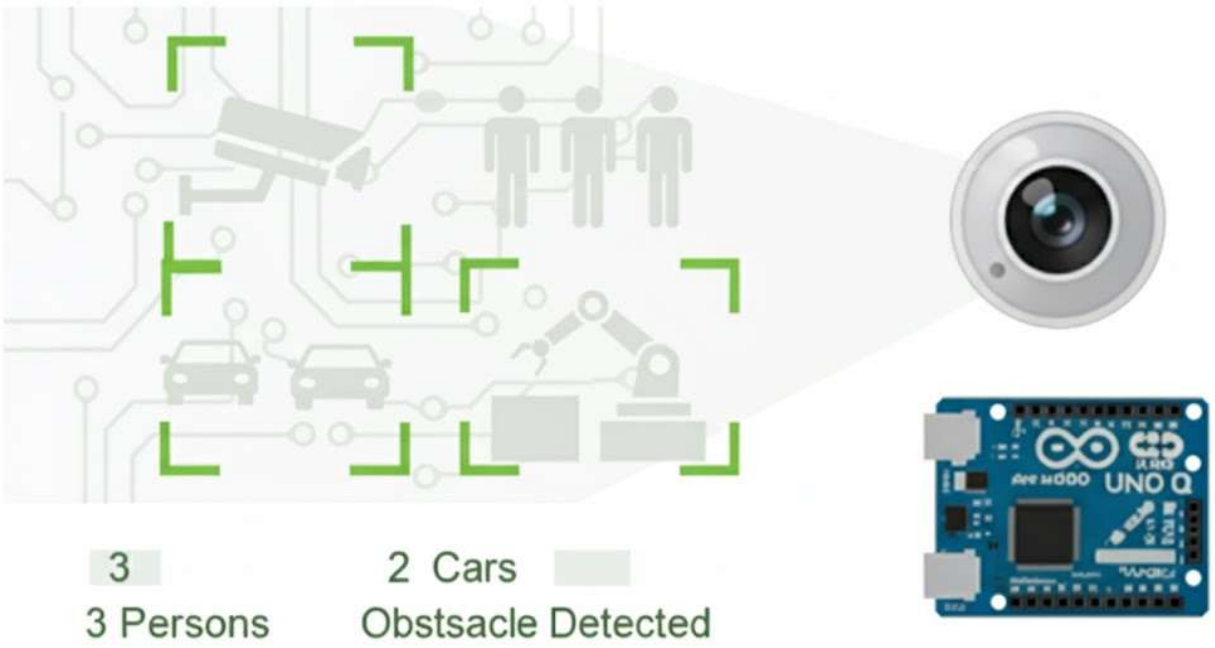
ก่อนถอดแฟลชไดรฟ์ ต้อง “unmount” ก่อนเสมอไม่ควรถอดแฟลชไดรฟ์ออกเลยทันที โดยเฉพาะใน Linux เพราะระบบจะ “หน่วงเวลาเขียนข้อมูล (write buffer)”

```
arduino@ThanaUNOQ:~$ sudo umount /media/usb
```

Object Detection

เหมาะสำหรับโปรเจกต์ที่ต้องการให้ Arduino UNO Q สามารถมองเห็นและระบุวัตถุในภาพ เช่น

- ระบบกล้องอัจฉริยะ
- การนับจำนวนคน/รถ
- หุ่นยนต์ที่ต้องหลบสิ่งกีดขวาง
- ยานยนต์ไร้คนขับ
- ระบบเฝ้าระวัง
- การเกษตรอัจฉริยะ



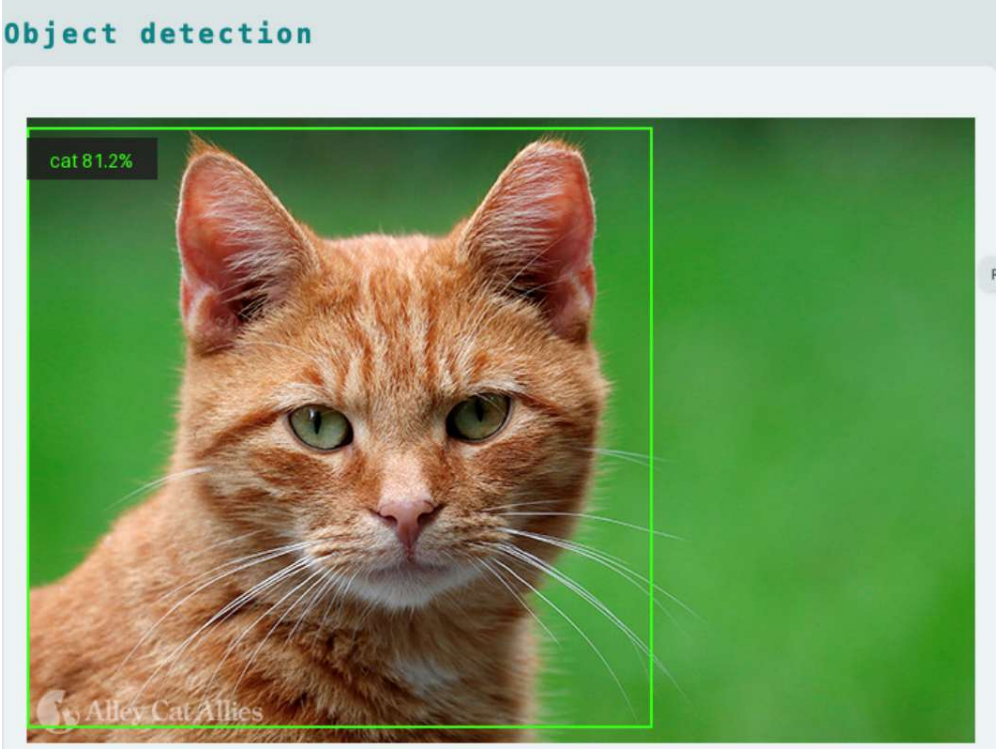
- **Object Detection (การตรวจจับวัตถุ):** เป็นงานด้านคอมพิวเตอร์วิทัศน์ (Computer Vision) ที่มีเป้าหมายในการระบุตำแหน่งของวัตถุหลายชนิดในภาพ พร้อมทั้งจัดประเภท (Classification) ให้กับวัตถุเหล่านั้น โดยมักแสดงผลเป็นกรอบสี่เหลี่ยมรอบวัตถุ (Bounding Box) และป้ายชื่อ (Label) พร้อมค่าความเชื่อมั่น (Confidence)
- **Model:** ในบริบทนี้คือ แบบจำลองปัญญาประดิษฐ์ (AI Model) ที่ได้รับการฝึกฝนด้วยข้อมูลจำนวนมาก จนสามารถเรียนรู้และทำนายผลลัพธ์บางอย่างได้ เช่น ในกรณีนี้คือการเรียนรู้ที่จะ "ตรวจจับ" ว่ามีวัตถุอะไรอยู่ในภาพและวัตถุนั้นอยู่ตรงไหน

Object Detection

ภาพรวม (Overview)

Object Detection Brick ช่วยให้สามารถ:

- ตรวจจับวัตถุในภาพได้ ทั้งจากไฟล์ภาพภายในเครื่องหรือจากกล้องโดยตรง
- ระบุตำแหน่งของวัตถุที่ตรวจจับได้ในภาพโดยใช้กรอบสี่เหลี่ยม (bounding boxes)
- รับค่าความมั่นใจ (confidence value) ของการตรวจจับแต่ละวัตถุและชื่อป้ายกำกับ (label)



Object Detection

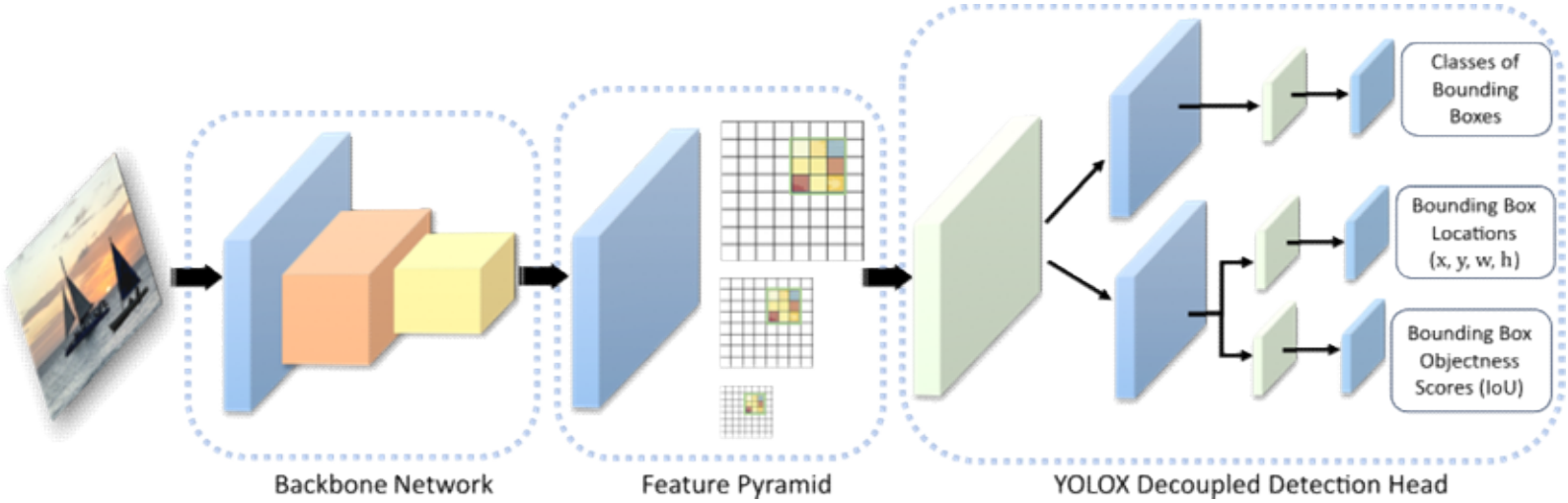
คุณสมบัติ (Features)

- ทำการตรวจจับวัตถุแบบเรียลไทม์บนภาพนิ่ง
- แสดงผลกรอบสี่เหลี่ยม, ป้ายชื่อวัตถุ (class label), และค่าความมั่นใจ (confidence score) สำหรับแต่ละวัตถุที่ตรวจพบ
- รองรับหลายรูปแบบไฟล์ภาพ เช่น JPEG, JPG, และ PNG (ค่าเริ่มต้น: JPG)
- อนุญาตให้ปรับค่าเกณฑ์ความมั่นใจในการตรวจจับและค่าการกรองซ้อน (Non-Maximum Suppression, NMS)
- สามารถเชื่อมต่อได้ง่ายกับภาพจากไลบรารี PIL หรือข้อมูลภาพแบบ byte stream โดยตรง

PIL ย่อมาจาก Python Imaging Library (ไลบรารีการประมวลผลภาพของ Python) เป็นไลบรารีพื้นฐานและเป็นที่นิยมอย่างมากสำหรับการจัดการกับไฟล์ภาพและประมวลผลภาพแบบแรสเตอร์ (Raster Images) ในภาษา Python

รูปภาพแรสเตอร์ (Raster Images) หรือที่มักเรียกกันว่า รูปภาพบิตแมป (Bitmap Images) คือประเภทของรูปภาพดิจิทัลที่ประกอบขึ้นจาก จุดสีเล็กๆ นับล้านจุด ที่เรียกว่า พิกเซล (Pixels)

YOLOX



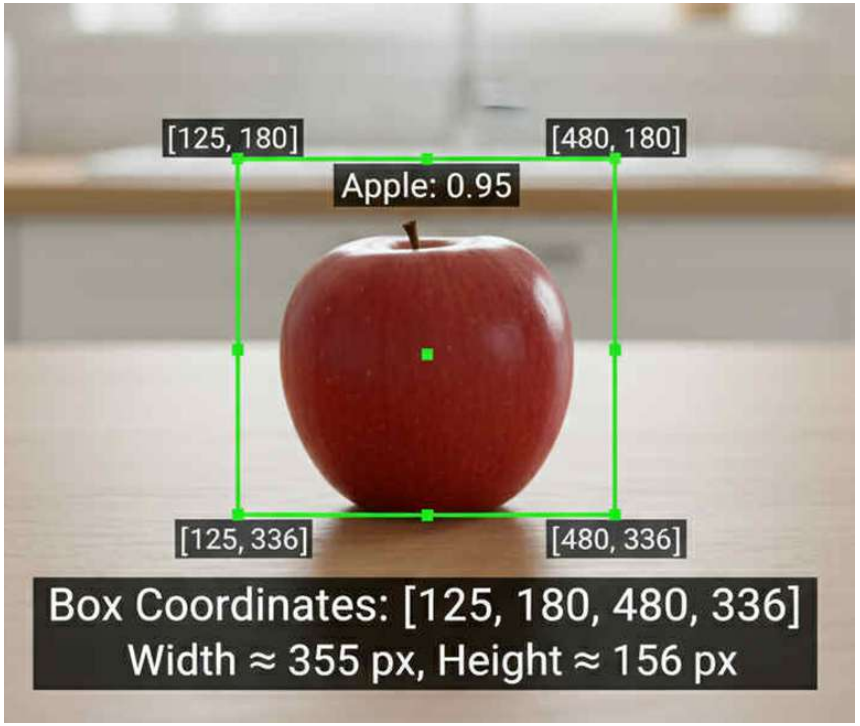
- แบบจำลองที่ใช้สำหรับการตรวจจับวัตถุของ Brick “Object Detection” จะใช้ YOLOX ที่เทรนมาจากชุดข้อมูล COCO
- โมเดล YOLO (You Only Look Once) เป็นหนึ่งในสถาปัตยกรรมโครงข่ายประสาทเทียมที่ได้รับความนิยมอย่างมากในการทำ Object Detection และ YOLOX ก็เป็นอีกหนึ่งเวอร์ชันที่ได้รับการปรับปรุงและพัฒนาขึ้นจากโมเดล YOLO
- YOLOX ย่อมาจาก You Only Look Once X เป็นแบบจำลอง (Model) ในตระกูล YOLO ที่ใช้สำหรับงาน การตรวจจับวัตถุแบบเรียลไทม์ (Real-time Object Detection) ในภาพและวิดีโอ โมเดลนี้ถูกพัฒนาขึ้นโดย Megvii ในปี 2021 โดยมีเป้าหมายเพื่อเพิ่มประสิทธิภาพด้านความแม่นยำและความเร็วให้เหนือกว่าโมเดล YOLO รุ่นก่อนหน้า (เช่น YOLOv3 และ YOLOv5

YOLOX ได้รวมเอาเทคนิคใหม่ๆ ที่ทำให้เป็นที่นิยมในวงการ Computer Vision ดังนี้:

คุณสมบัติ	คำอธิบาย
Anchor-Free	แตกต่างจาก YOLO รุ่นก่อนๆ YOLOX ไม่ใช้ "Anchor Boxes" (กรอบที่กำหนดขนาดล่วงหน้า) แต่จะทำนายตำแหน่งวัตถุโดยตรงจากจุดศูนย์กลาง ทำให้ **การออกแบบโมเดลง่ายขึ้น** และ **มีความยืดหยุ่นสูง**
Decoupled Head	ส่วนที่ใช้ทำนายผลลัพธ์ (Head) จะถูกแยกออกเป็นสามส่วนหลักสำหรับการทำนาย **คลาสของวัตถุ**, **ตำแหน่งของกรอบ** (Bounding Box), และ **คะแนนความเชื่อมั่น** (Confidence Score) ซึ่งช่วยให้โมเดลมี **ประสิทธิภาพและความแม่นยำในการทำนายสูงขึ้น**
SimOTA (Simplified Optimal Transport Assignment)	เป็นกลยุทธ์การกำหนดป้ายกำกับ (Label Assignment) แบบใหม่ที่ชาญฉลาด เพื่อให้โมเดลเรียนรู้ว่าควรจับคู่ผลทำนายใดกับวัตถุจริงในภาพ ช่วยให้การฝึก **แม่นยำและเสถียรยิ่งขึ้น**

ตำแหน่งของกรอบ (Bounding Box)

- Bounding Box คือ ตำแหน่งของกรอบสี่เหลี่ยมของวัตถุที่ตรวจจับได้ ซึ่งเป็น พิกัดสี่เหลี่ยมล้อมรอบวัตถุ
- จะอยู่ในรูปแบบ [x_min, y_min, x_max, y_max] โดย x_min คือ พิกัดแนวนอนซ้ายสุดของกล่อง, y_min คือ พิกัดแนวตั้งบนสุดของกล่อง, x_max คือ พิกัดแนวนอนขวาสุดของกล่อง, และ y_max คือ พิกัดแนวตั้งล่างสุดของกล่อง



หมายความว่า:

- ซ้ายสุดของวัตถุอยู่ที่ x = 125
- บนสุดของวัตถุอยู่ที่ y = 180
- ขวาสุดของวัตถุอยู่ที่ x = 480
- ล่างสุดของวัตถุอยู่ที่ y = 336

ดังนั้นกล่องล้อมรอบวัตถุในภาพจะมี **กว้าง** ≈ 480 – 125 ≈ 355 px และ **สูง** ≈ 336 – 180 ≈ 156 px

ตารางคลาส YOLOX (COCO Dataset – 80 Classes)

ลำดับ	Class (English)	คำแปลภาษาไทย
1	person	คน
2	bicycle	จักรยาน
3	car	รถยนต์
4	motorcycle	รถจักรยานยนต์
5	airplane	เครื่องบิน
6	bus	รถบัส
7	train	รถไฟ
8	truck	รถบรรทุก
9	boat	เรือ
10	traffic light	สัญญาณไฟจราจร
11	fire hydrant	หัวจ่ายน้ำดับเพลิง
12	stop sign	ป้ายหยุด
13	parking meter	มิเตอร์จอดรถ
14	bench	ม้านั่ง
15	bird	นก
16	cat	แมว
17	dog	สุนัข
18	horse	ม้า
19	sheep	แกะ
20	cow	วัว
21	elephant	ช้าง
22	bear	หมี
23	zebra	ม้าลาย
24	giraffe	ยีราฟ
25	backpack	กระเป๋าเป้
26	umbrella	ร่ม
27	handbag	กระเป๋าถือ
28	tie	เนกไท
29	suitcase	กระเป๋าเดินทาง
30	frisbee	จานร่อน
31	skis	สกี
32	snowboard	สโนว์บอร์ด
33	sports ball	ลูกบอลกีฬา
34	kite	ว่าว
35	baseball bat	ไม้เบสบอล
36	baseball glove	ถุงมือเบสบอล
37	skateboard	สเก็ตบอร์ด
38	surfboard	กระดานโต้คลื่น
39	tennis racket	ไม้เทนนิส
40	bottle	ขวด
41	wine glass	แก้วไวน์
42	cup	ถ้วย
43	fork	ส้อม
44	knife	มีด
45	spoon	ช้อน
46	bowl	ชาม
47	banana	กล้วย
48	apple	แอปเปิ้ล
49	sandwich	แซนด์วิช
50	orange	ส้ม
51	broccoli	บร็อกโคลี
52	carrot	แครอท
53	hot dog	ฮอตด็อก
54	pizza	พิซซ่า
55	donut	โดนัท
56	cake	เค้ก
57	chair	เก้าอี้
58	couch	โซฟา
59	potted plant	กระถางต้นไม้
60	bed	เตียง
61	dining table	โต๊ะอาหาร
62	toilet	โถสุขภัณฑ์
63	tv	โทรทัศน์
64	laptop	แล็ปท็อป
65	mouse	เมาส์
66	remote	รีโมต
67	keyboard	แป้นพิมพ์
68	cell phone	โทรศัพท์มือถือ
69	microwave	ไมโครเวฟ
70	oven	เตาอบ
71	toaster	เครื่องปิ้งขนมปัง
72	sink	อ่างล้างจาน
73	refrigerator	ตู้เย็น
74	book	หนังสือ
75	clock	นาฬิกา
76	vase	แจกัน
77	scissors	กรรไกร
78	teddy bear	ตุ๊กตาหมี
79	hair drier	ไดร์เป่าผม
80	toothbrush	แปรงสีฟัน

```
import os
from arduino.app_bricks.object_detection import
ObjectDetection
from arduino.app_utils import *
import time

def main():
    img_path = "Man.jpg"
    if not os.path.exists(img_path):
        print(f“ไม่เจอรูปภาพนะครับเจ้านาย: {img_path}")
        return
    print("เจอรูปภาพแล้ว ทำงานต่อได้")
    object_detection = ObjectDetection()

with open(img_path, "rb") as f:
        frame = f.read()
    out = object_detection.detect(frame)
    if out and "detection" in out:
        for i, obj_det in enumerate(out["detection"]):
            print(f"[{i+1}] Object: {obj_det.get('class_name')}, Confidence:
{obj_det.get('confidence')}")
            print(f"Bounding box: {obj_det.get('bounding_box_xyxy')}\n")
    out_image = object_detection.draw_bounding_boxes(frame, out)
    print("ตรวสอบภาพเสร็จละครับเจ้านาย")
    time.sleep(2)

if _name_ == "_main_":
    main()
```

การรัน App บน Arduino App Lab ในครั้งแรกอาจจะมีการดาวน์โหลดไฟล์ต่าง ๆ เข้ามาไว้ที่ UNO Q

```
App launch
fe5dceedfb61 Downloading [==========================>]  50.81MB/89.07MB
408ff3d07603 Downloading [=====================>]  60.01MB/137.5MB
fe5dceedfb61 Downloading [==========================>]  51.34MB/89.07MB
529ec9b441f0 Downloading [======================>]  29.74MB/63.9MB
fe5dceedfb61 Downloading [==========================>]  51.88MB/89.07MB
408ff3d07603 Downloading [=====================>]  60.55MB/137.5MB
529ec9b441f0 Downloading [======================>]  30.28MB/63.9MB
408ff3d07603 Downloading [=====================>]  61.09MB/137.5MB
fe5dceedfb61 Downloading [==========================>]  52.42MB/89.07MB
fe5dceedfb61 Downloading [==========================>]  52.95MB/89.07MB
408ff3d07603 Downloading [=====================>]  61.64MB/137.5MB
529ec9b441f0 Downloading [======================>]  30.82MB/63.9MB
fe5dceedfb61 Downloading [==========================>]  53.48MB/89.07MB
408ff3d07603 Downloading [=====================>]  62.18MB/137.5MB
529ec9b441f0 Downloading [======================>]  31.36MB/63.9MB
fe5dceedfb61 Downloading [==========================>]  54.02MB/89.07MB
408ff3d07603 Downloading [=====================>]  62.72MB/137.5MB
fe5dceedfb61 Downloading [==========================>]  54.56MB/89.07MB
529ec9b441f0 Downloading [======================>]   31.9MB/63.9MB
fe5dceedfb61 Downloading [==========================>]   55.1MB/89.07MB
408ff3d07603 Downloading [=====================>]  63.26MB/137.5MB
529ec9b441f0 Downloading [======================>]  32.44MB/63.9MB
fe5dceedfb61 Downloading [==========================>]  55.64MB/89.07MB
408ff3d07603 Downloading [=====================>]   63.8MB/137.5MB
fe5dceedfb61 Downloading [==========================>]  56.17MB/89.07MB
408ff3d07603 Downloading [=====================>]  64.34MB/137.5MB
529ec9b441f0 Downloading [======================>]  32.98MB/63.9MB
fe5dceedfb61 Downloading [==========================>]  56.71MB/89.07MB
529ec9b441f0 Downloading [======================>]  33.52MB/63.9MB
408ff3d07603 Downloading [=====================>]  64.88MB/137.5MB
fe5dceedfb61 Downloading [==========================>]  57.25MB/89.07MB
408ff3d07603 Downloading [=====================>]  65.42MB/137.5MB
529ec9b441f0 Downloading [======================>]  34.06MB/63.9MB
fe5dceedfb61 Downloading [==========================>]  57.78MB/89.07MB
408ff3d07603 Downloading [=====================>]  65.96MB/137.5MB
fe5dceedfb61 Downloading [==========================>]  58.31MB/89.07MB
529ec9b441f0 Downloading [======================>]   34.6MB/63.9MB
408ff3d07603 Downloading [=====================>]   66.5MB/137.5MB
fe5dceedfb61 Downloading [==========================>]  58.85MB/89.07MB
fe5dceedfb61 Downloading [==========================>]  59.38MB/89.07MB
408ff3d07603 Downloading [=====================>]  67.04MB/137.5MB
529ec9b441f0 Downloading [======================>]  35.14MB/63.9MB
fe5dceedfb61 Downloading [==========================>]  59.92MB/89.07MB
408ff3d07603 Downloading [=====================>]  67.58MB/137.5MB
529ec9b441f0 Downloading [======================>]  35.68MB/63.9MB
fe5dceedfb61 Downloading [==========================>]  60.46MB/89.07MB
408ff3d07603 Downloading [=====================>]  68.12MB/137.5MB
529ec9b441f0 Downloading [======================>]  36.22MB/63.9MB
fe5dceedfb61 Downloading [==========================>]     61MB/89.07MB
408ff3d07603 Downloading [=====================>]  68.66MB/137.5MB
fe5dceedfb61 Downloading [==========================>]  61.54MB/89.07MB
529ec9b441f0 Downloading [======================>]  36.76MB/63.9MB
```

เอาต์พุตที่ได้

```
✅ Object detection complete.
Activating python virtual environment
2025-11-13 07:06:10.765 WARNING - [MainThread] arduino.app_internal.core.ei:  [ObjectDetection] Host: ei-obj-detection-runner - Ports: 1337 - URL: http://ei-obj-detection-runner:1337
[1] Object: car, Confidence: 81.04
Bounding box: [125.48076923076923, 179.80769230769232, 480.28846153846155, 335.5769230769231]

✅ Object detection complete.
exited with code 0
```

```
✅ Object detection complete.
2025-11-13 06:43:11.106 WARNING - [MainThread] arduino.app_internal.core.ei:  [ObjectDetection] Host: ei-obj-detection-runner - Ports: 1337 - URL: http://ei-obj-detection-runner:1337
[1] Object: person, Confidence: 94.51
Bounding box: [445.53846153846155, 77.30769230769232, 846.7692307692307, 961.0]

[2] Object: tie, Confidence: 80.08
Bounding box: [576.0, 348.0769230769231, 652.3076923076923, 761.6153846153846]

✅ Object detection complete.
```

```
Activating python virtual environment
2025-11-13 07:02:39.764 WARNING - [MainThread] arduino.app_internal.core.ei:  [ObjectDetection] Host: ei-obj-detection-runner - Ports: 1337 - URL: http://ei-obj-detection-runner:1337
[1] Object: dog, Confidence: 65.96
Bounding box: [2604.4615384615386, 859.846153846154, 3925.3846153846152, 3314.7692307692314]

[2] Object: dog, Confidence: 59.92
Bounding box: [1345.846153846154, 996.9230769230769, 2729.0769230769233, 3202.6153846153848]

✅ Object detection complete.
exited with code 0
```

Python จะส่ง `frame` ไปที่ `http://ei-obj-detection-runner:1337` เพื่อให้ service ตรวจจับวัตถุ แล้วคืนผลกลับมา