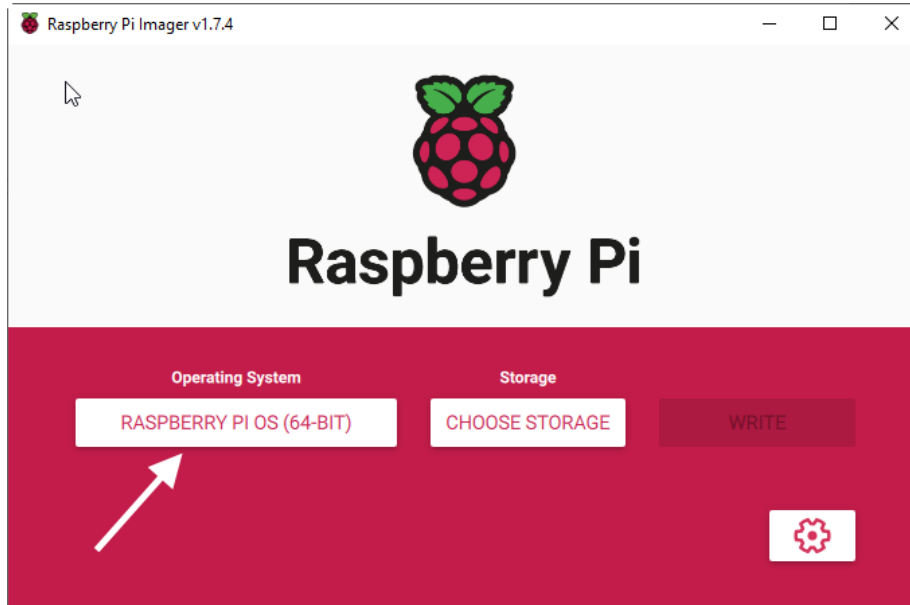



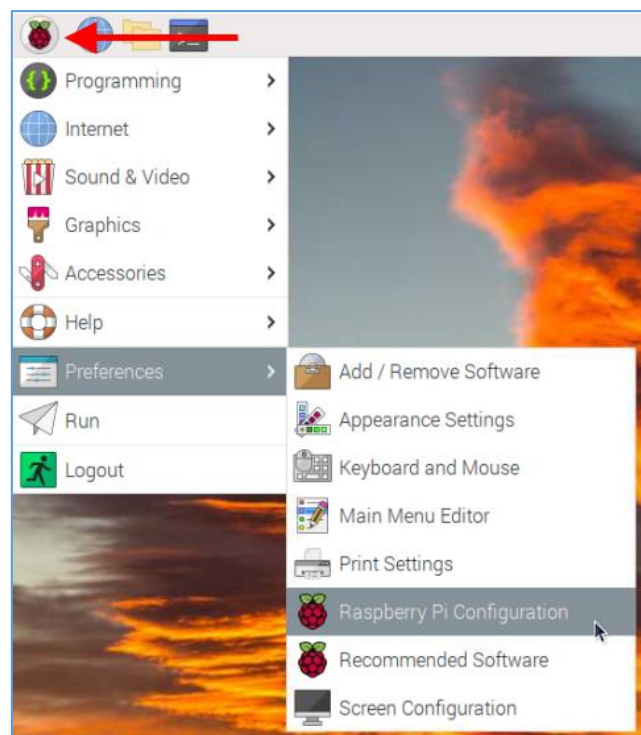
## Machine Learning for Raspberry Pi (64 บิต)

### 1. ติดตั้งระบบปฏิบัติการ Raspbian 64 บิต

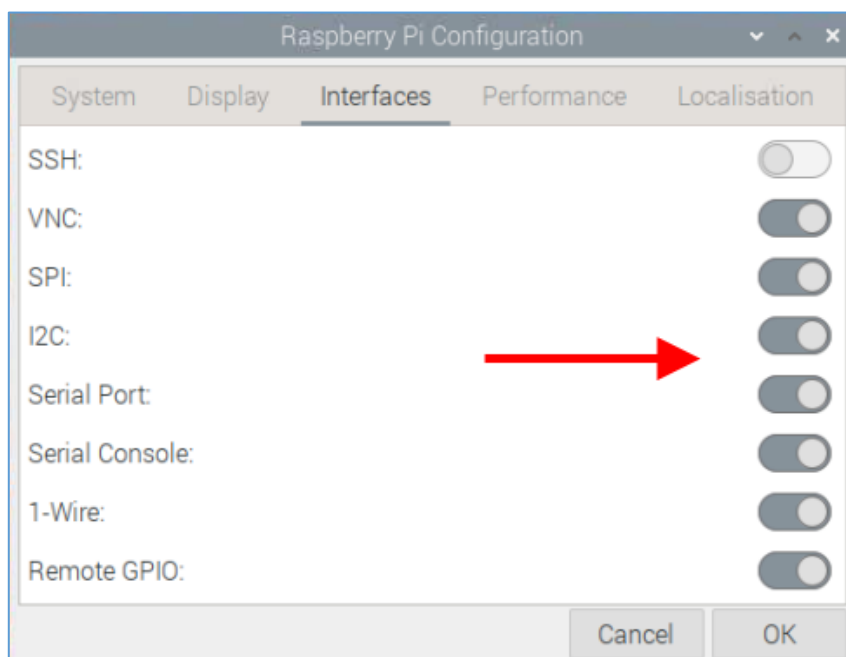


### 2. เปิดใช้งานอินเทอร์เน็ตเฟส

2.1 คลิกที่  --> Preferences --> Raspberry Pi Configuration

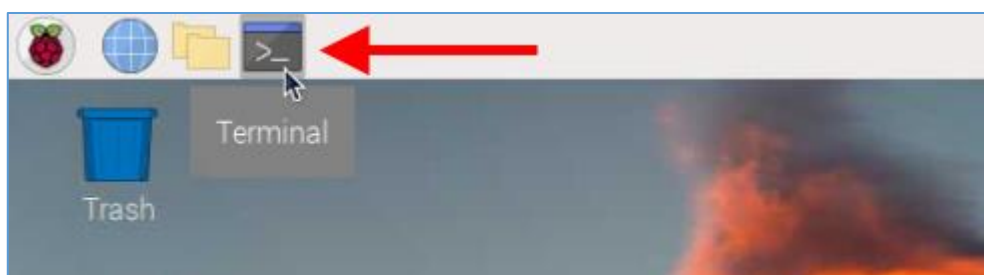


2.2 คลิกเปิดการใช้งาน แล้วคลิกปุ่ม OK

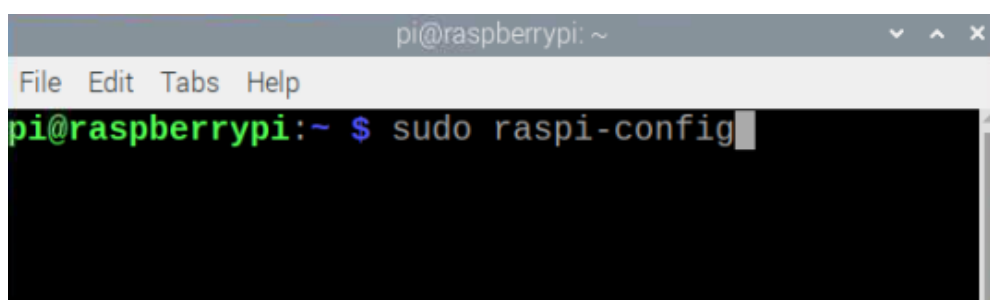


### 3. เปิดใช้งานกล้อง

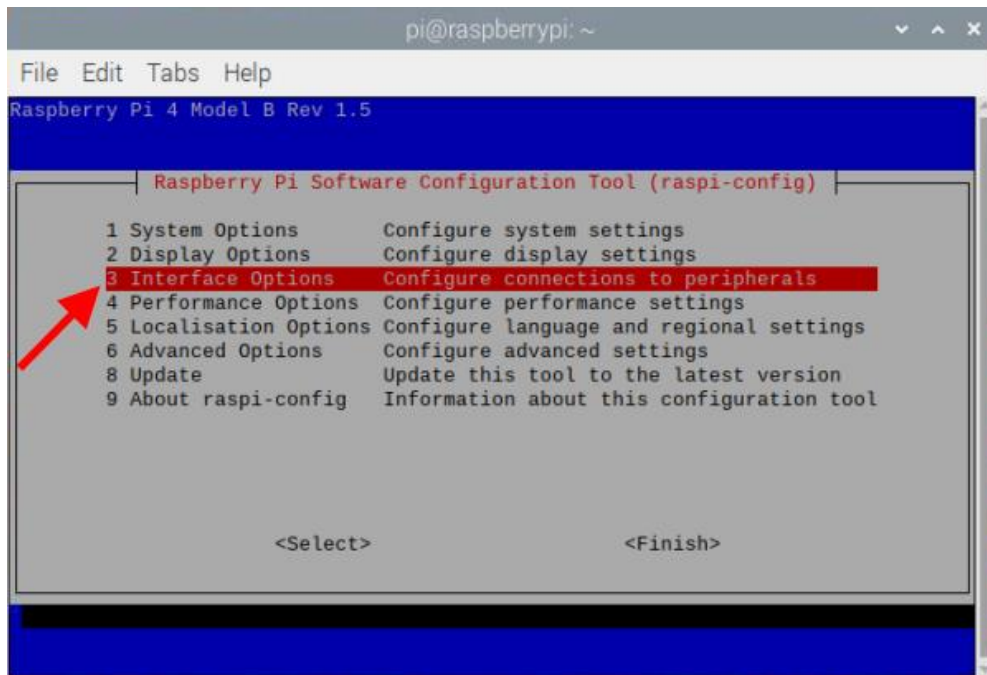
3.1 เปิดโปรแกรม Terminal



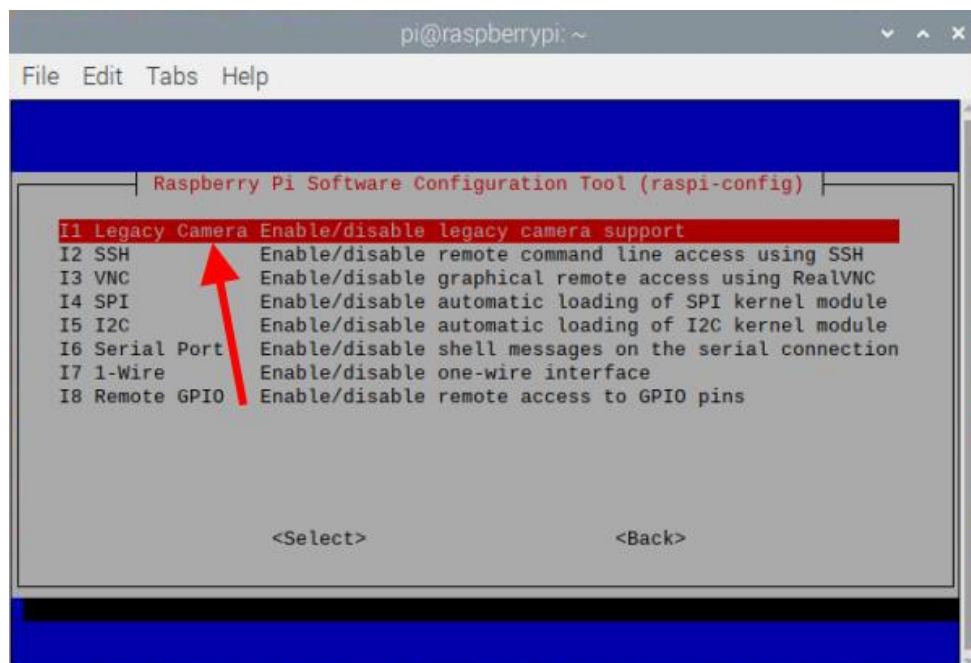
3.2 พิมพ์คำสั่ง `sudo raspi-config`



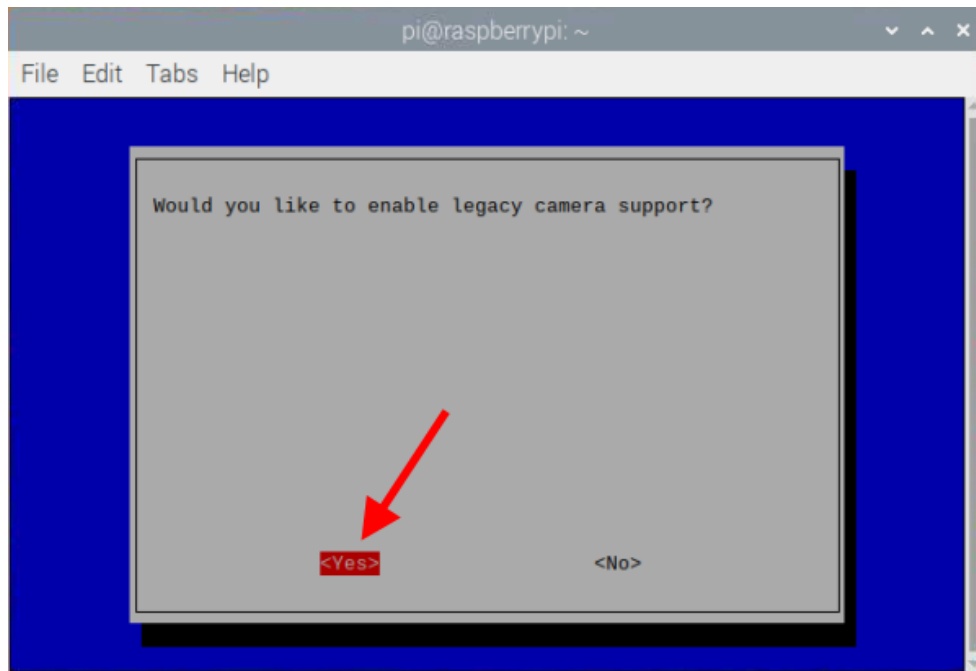
### 3.3 เลือก Interface Options แล้ว Enter



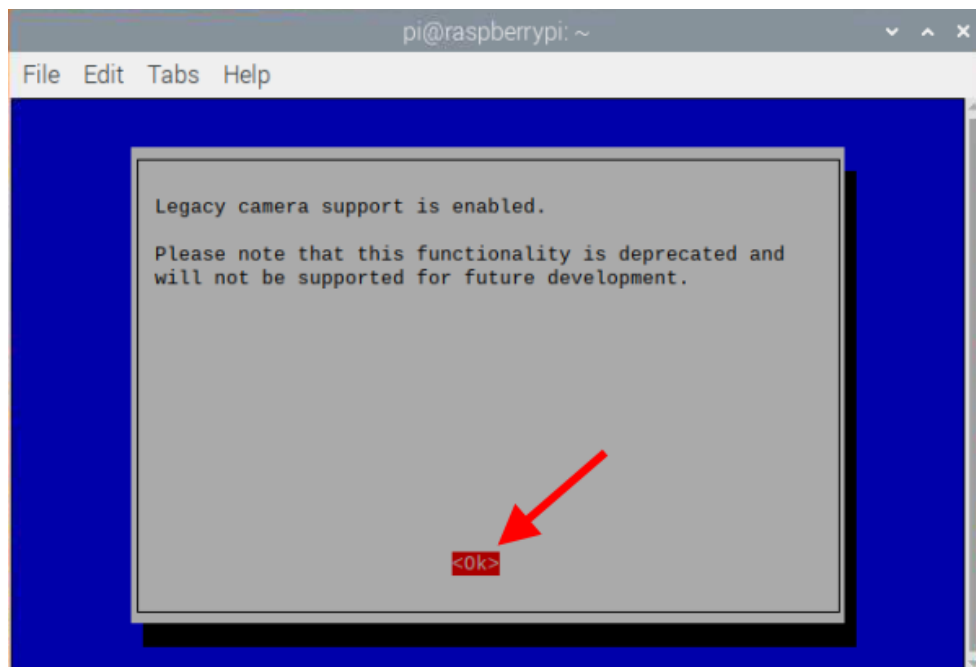
### 3.4 เลือก I1 Legacy Camera แล้ว Enter



### 3.5 เลือก <Yes> แล้ว Enter

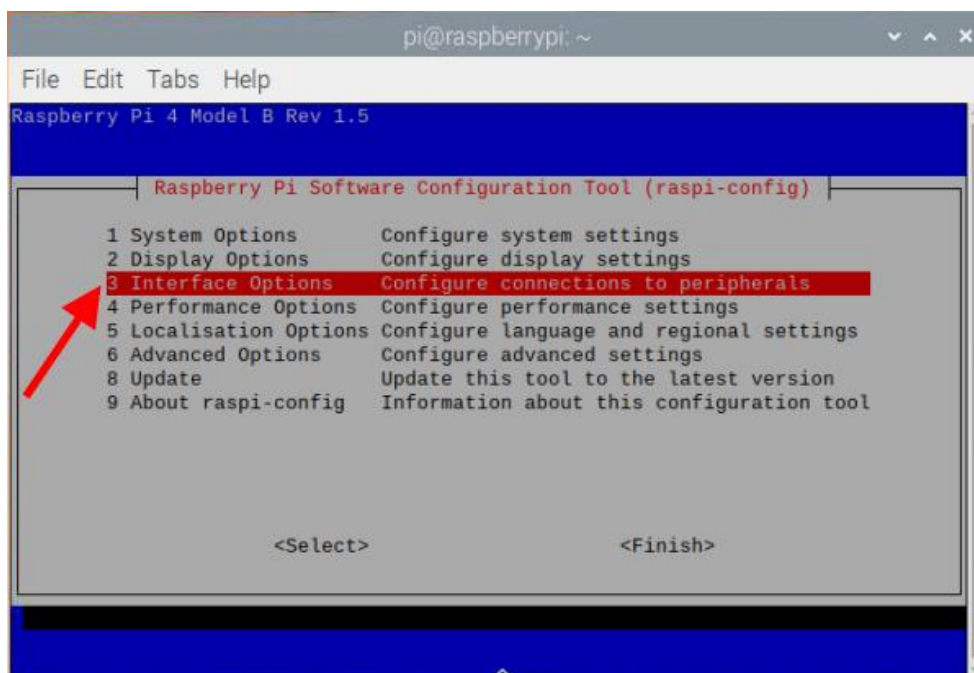


### 3.6 เลือก <Ok> แล้ว Enter

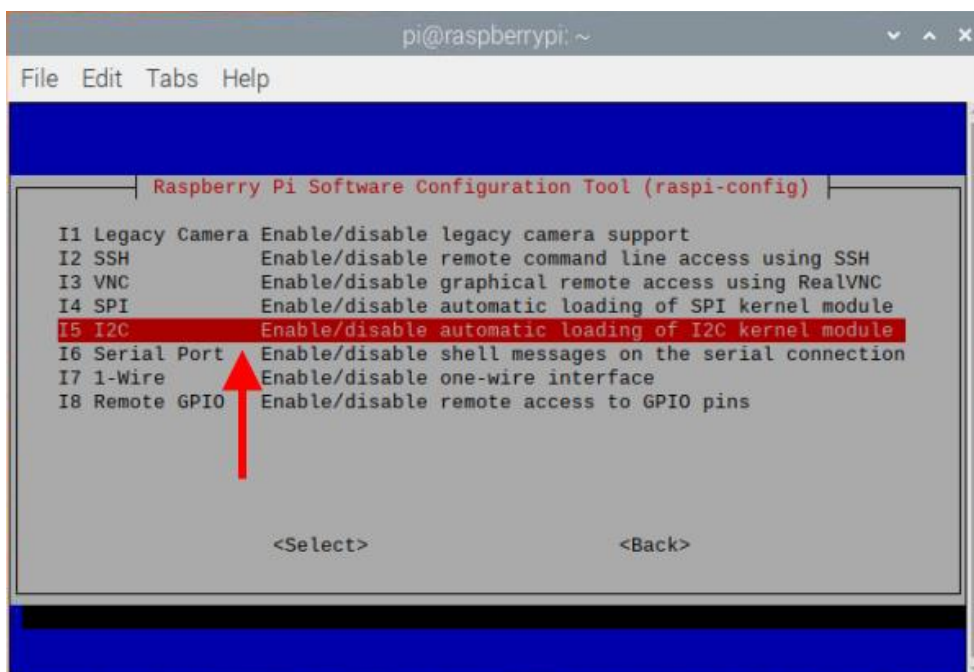




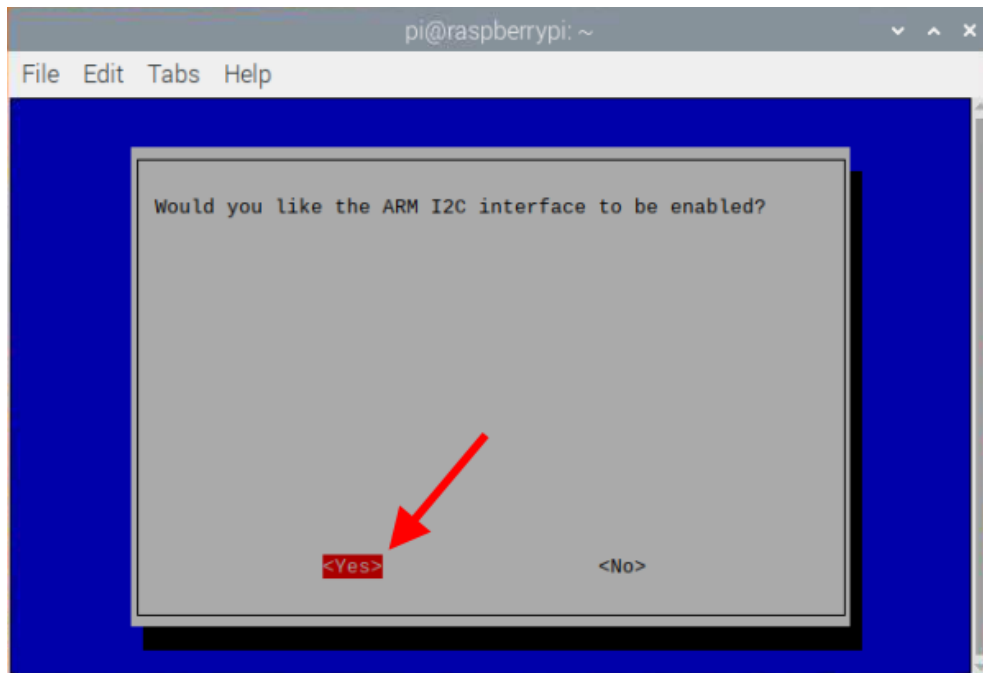
3.7 เลือก Interface Options อีกครั้ง แล้ว Enter



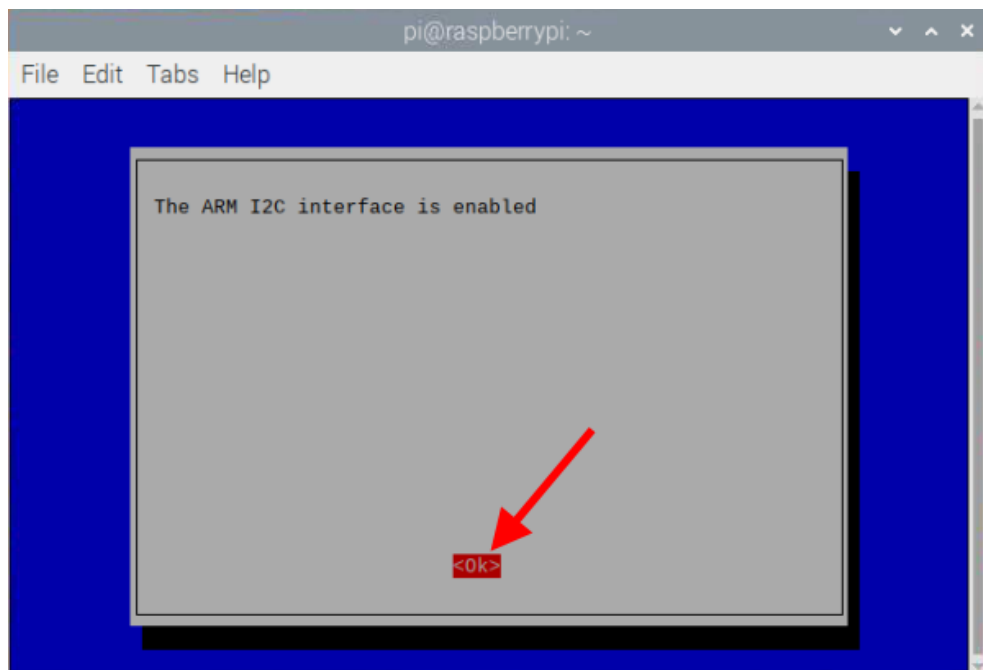
3.8 เปิดใช้งาน I2C โดยเลือก I5 I2C แล้ว Enter



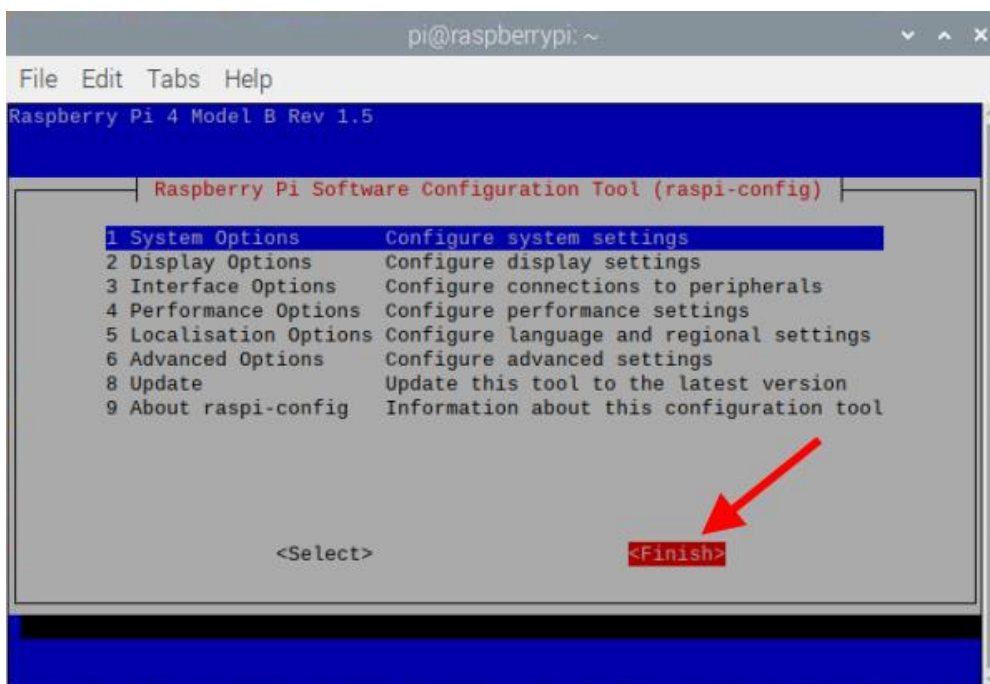
### 3.9 เลือก <Yes> แล้ว Enter



### 3.10 เลือก <Ok> แล้ว Enter



### 3.11 เลือก <Finish> แล้ว Enter



### 3.12 ระบบจะให้ Reboot 1 ครั้ง

## 4. ติดตั้งไลบรารี OpenCV

### 4.1 พิมพ์คำสั่ง `sudo apt install python3-opencv`

```

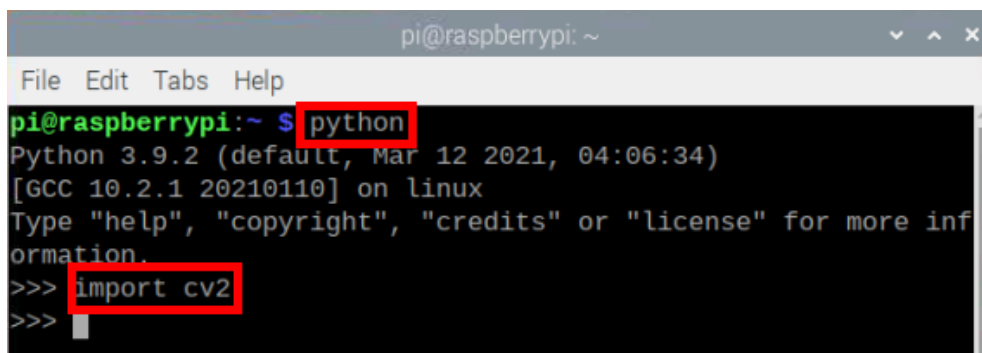
pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ sudo apt install python3-opencv
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and is no longer required:
  libfuse2
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  gdal-data libaec0 libarmadillo10 libarpac2 libcfitsio9 libcharls2 libdap27
  libdapclient6v5 libepsilon1 libfreexl1 libfyba0 libgdal28 libgdcm3.0 libgeos-3.9.0
  libgeos-c1v5 libgeotiff5 libhdf4-0-alt libhdf5-103-1 libhdf5-hl-100 libheif1 libkmlba
  se1
  libkmldev1 libkmlengine1 libleft5 libmariadb3 libminizip1 libnetcdf18 libodbc1
  libogdi4.1 libopencv-calib3d4.5 libopencv-contrib4.5 libopencv-core4.5 libopencv-dnn4
  .5
  libopencv-features2d4.5 libopencv-flann4.5 libopencv-highgui4.5 libopencv-imgcodecs4.
  5
  libopencv-imgproc4.5 libopencv-ml4.5 libopencv-objdetect4.5 libopencv-photo4.5
  libopencv-shape4.5 libopencv-stitching4.5 libopencv-video4.5 libopencv-videoio4.5 lib
  pq5

```

#### 4.2 ทดสอบโดยพิมพ์คำสั่ง

`python`

`import cv2`



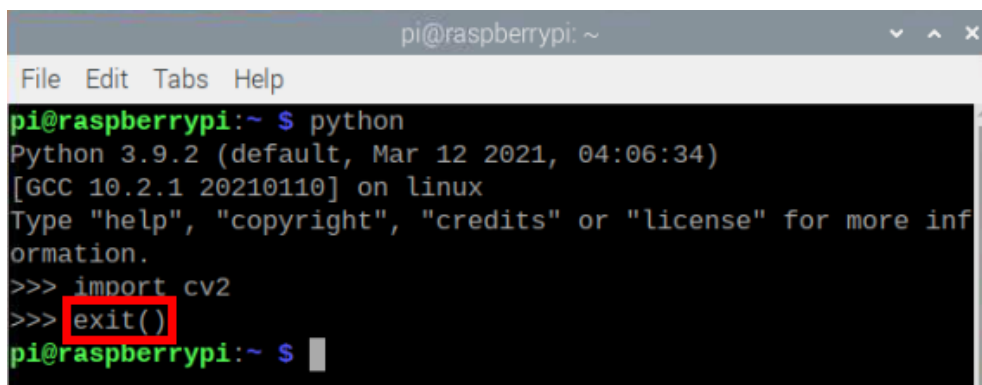
```

pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ python
Python 3.9.2 (default, Mar 12 2021, 04:06:34)
[GCC 10.2.1 20210110] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import cv2
>>>

```

#### 4.3 ตรวจสอบเวอร์ชัน พิมพ์คำสั่ง `cv2.__version__`

#### 4.4 ออกจากคำสั่ง python โดยพิมพ์คำสั่ง `exit()`

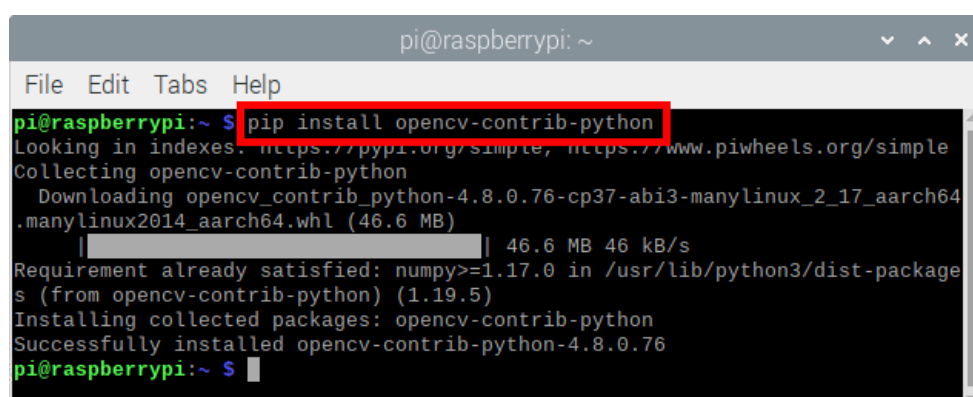


```

pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ python
Python 3.9.2 (default, Mar 12 2021, 04:06:34)
[GCC 10.2.1 20210110] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import cv2
>>> exit()
pi@raspberrypi:~ $

```

#### 4.5 พิมพ์คำสั่ง `pip install opencv-contrib-python`



```

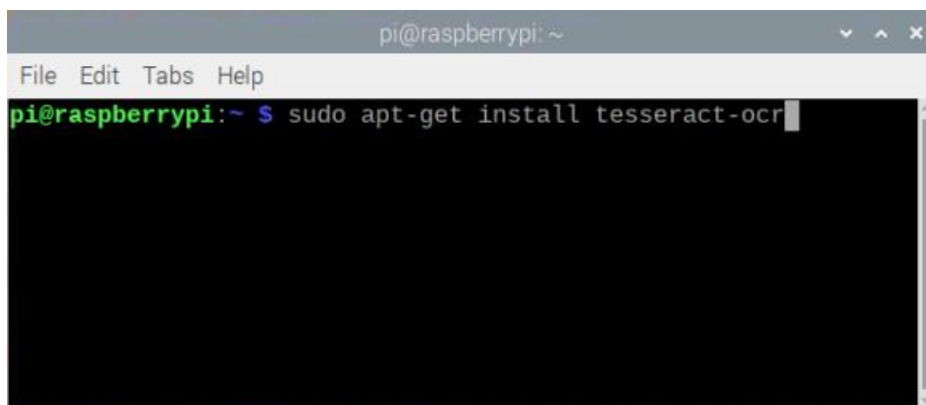
pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ pip install opencv-contrib-python
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Collecting opencv-contrib-python
  Downloading opencv_contrib_python-4.8.0.76-cp37-abi3-manylinux_2_17_aarch64
  .manylinux2014_aarch64.whl (46.6 MB)
    | 46.6 MB 46 kB/s
Requirement already satisfied: numpy>=1.17.0 in /usr/lib/python3/dist-packages
 (from opencv-contrib-python) (1.19.5)
Installing collected packages: opencv-contrib-python
Successfully installed opencv-contrib-python-4.8.0.76
pi@raspberrypi:~ $

```

## Workshop 1 Text Detection การแปลงข้อความที่อยู่ในรูปภาพให้เป็นข้อความ

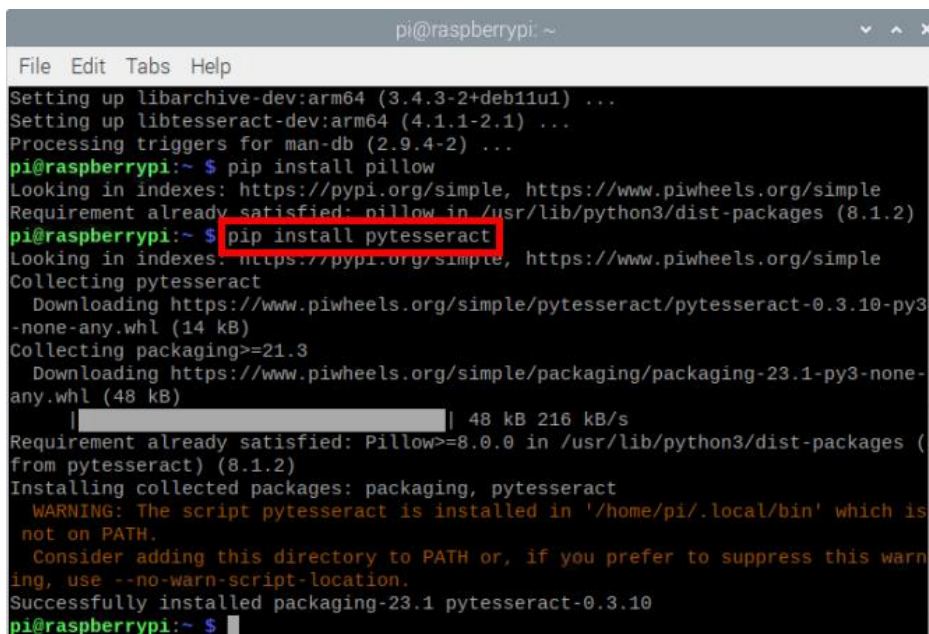
### 1. ติดตั้งไลบรารี pytesseract

1.1 เปิดโปรแกรม Terminal จากนั้นพิมพ์คำสั่ง `sudo apt install tesseract-ocr`



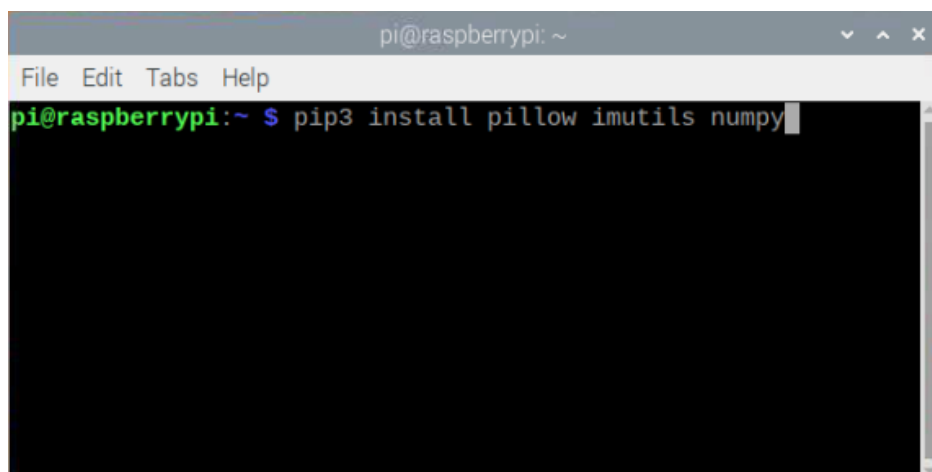
```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~ $ sudo apt-get install tesseract-ocr
```

1.2 พิมพ์คำสั่ง `pip install pytesseract`



```
pi@raspberrypi: ~  
File Edit Tabs Help  
Setting up libarchive-dev:arm64 (3.4.3-2+deb11u1) ...  
Setting up libtesseract-dev:arm64 (4.1.1-2.1) ...  
Processing triggers for man-db (2.9.4-2) ...  
pi@raspberrypi:~ $ pip install pillow  
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple  
Requirement already satisfied: pillow in /usr/lib/python3/dist-packages (8.1.2)  
pi@raspberrypi:~ $ pip install pytesseract  
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple  
Collecting pytesseract  
  Downloading https://www.piwheels.org/simple/pytesseract/pytesseract-0.3.10-py3  
-none-any.whl (14 kB)  
Collecting packaging>=21.3  
  Downloading https://www.piwheels.org/simple/packaging/packaging-23.1-py3-none-  
any.whl (48 kB)  
    |████████████████████████████████████████| 48 kB 216 kB/s  
Requirement already satisfied: Pillow>=8.0.0 in /usr/lib/python3/dist-packages (  
from pytesseract) (8.1.2)  
Installing collected packages: packaging, pytesseract  
  WARNING: The script pytesseract is installed in '/home/pi/.local/bin' which is  
  not on PATH.  
  Consider adding this directory to PATH or, if you prefer to suppress this warn  
ing, use --no-warn-script-location.  
Successfully installed packaging-23.1 pytesseract-0.3.10  
pi@raspberrypi:~ $
```

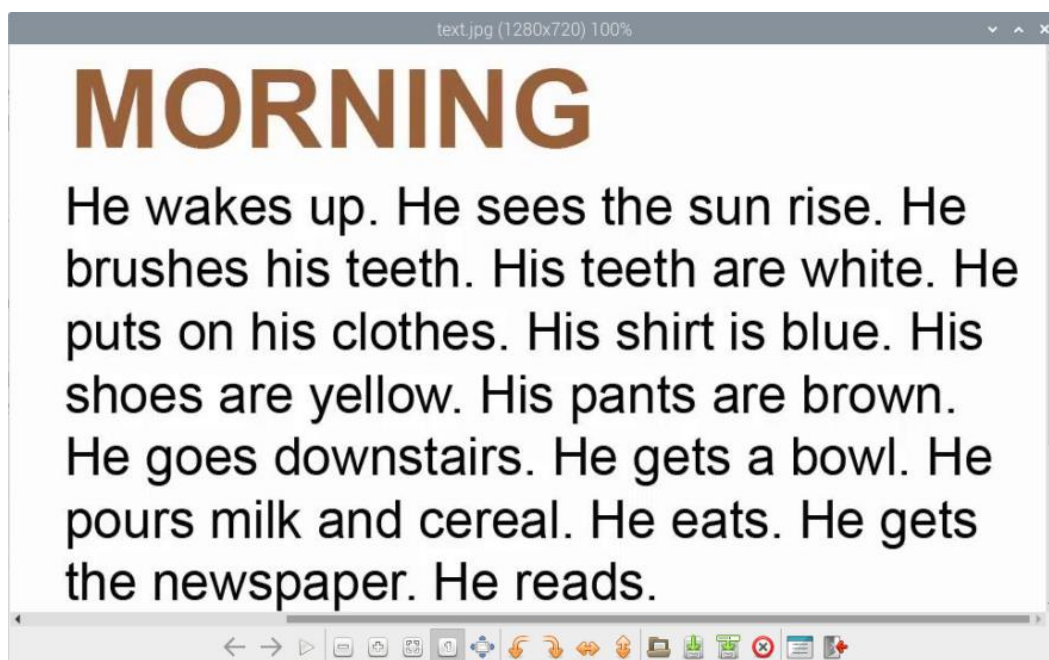
2. ติดตั้ง pillow, imutils และ numpy โดยพิมพ์คำสั่ง `pip3 install pillow imutils numpy`



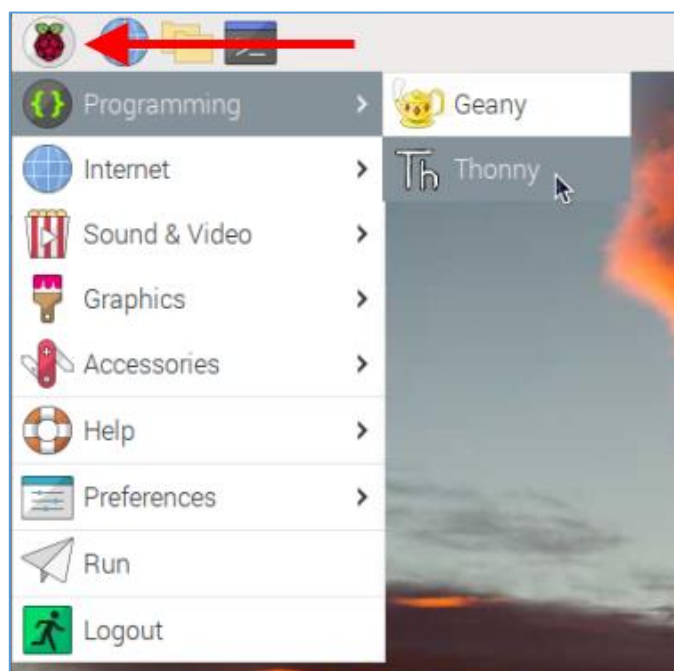
```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~ $ pip3 install pillow imutils numpy
```

3. Workshop 1.1 โปรแกรมแปลงข้อความจากรูปภาพให้เป็นข้อความ

3.1 ค้นหารูปภาพที่มีข้อความจากอินเทอร์เน็ต บันทึกในโฟลเดอร์ /home/pi โดยตั้งชื่อว่า text.jpg ตัวอย่างดังรูป



3.2 เปิดโปรแกรม Thonny โดยคลิกที่  --> Programming --> Thonny

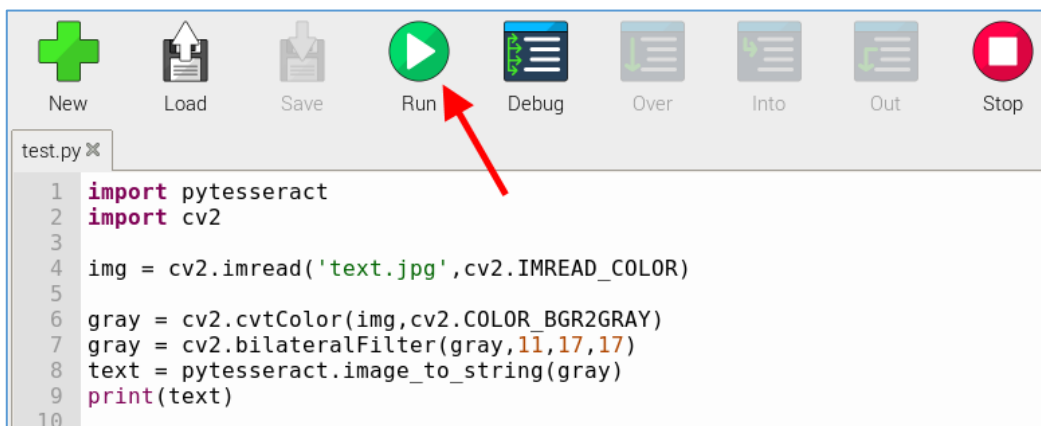


3.3 เขียนโค้ดคำสั่ง ดังรูป

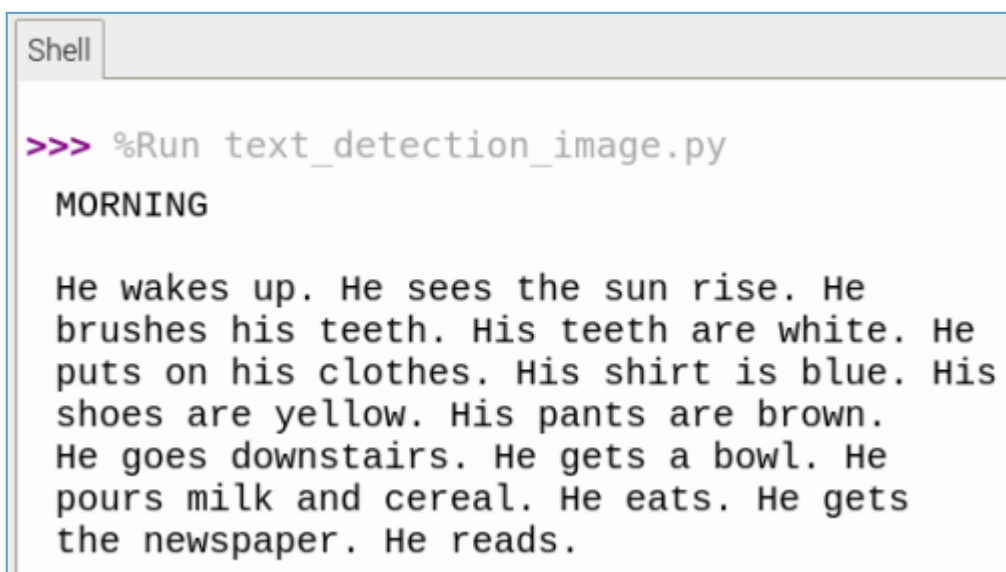
```
test.py ✕  
1 import pytesseract  
2 import cv2  
3  
4 img = cv2.imread('text.jpg',cv2.IMREAD_COLOR)  
5  
6 gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)  
7 gray = cv2.bilateralFilter(gray,11,17,17)  
8 text = pytesseract.image_to_string(gray)  
9 print(text)
```



### 3.4 คลิกปุ่ม Run



### 3.5 จะได้อ่านข้อความที่อยู่ในรูปภาพ text.jpg ปรากฏขึ้นมาในช่อง Shell



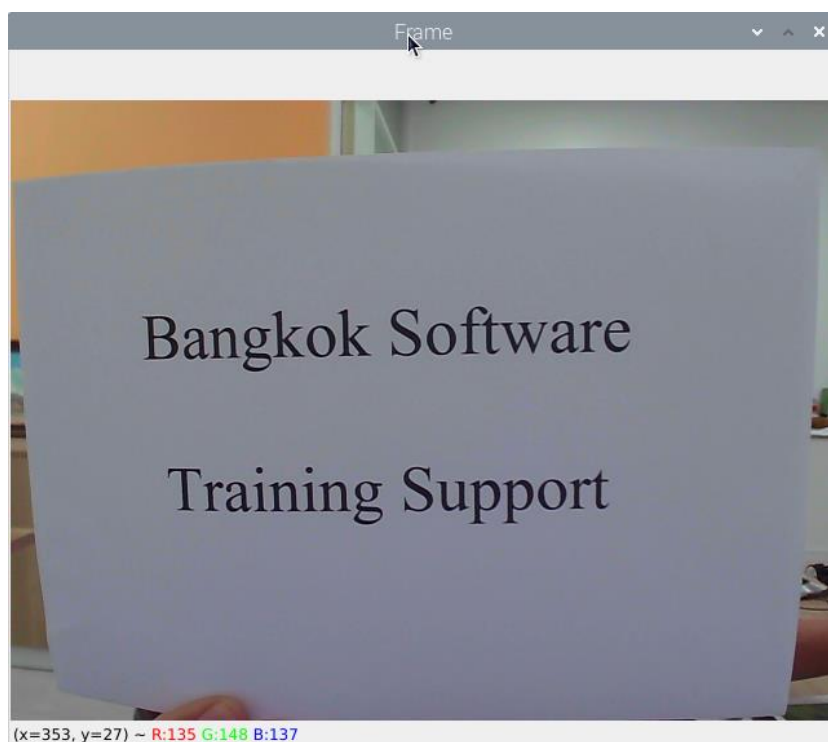


#### 4. Workshop 1.2 โปรแกรมแปลงข้อความจากรูปภาพที่ส่งผ่านกล้องเว็บแคมให้เป็นข้อความ แบบที่ 1

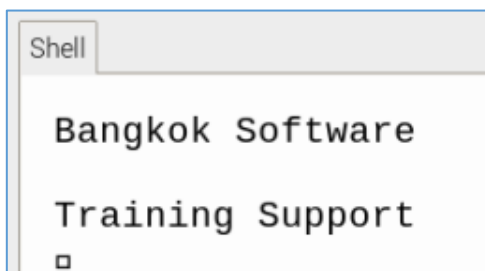
##### 4.1 เขียนโค้ดคำสั่ง ดังรูป

```
test2.py x
1 import pytesseract
2 import cv2
3
4 cam = cv2.VideoCapture(0)
5
6 while True:
7     img, frame = cam.read()
8     gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
9     gray = cv2.bilateralFilter(gray, 11, 17, 17)
10    cv2.imshow('Show', frame)
11
12    key = cv2.waitKey(1) & 0xFF
13    if key == ord('s'):
14        text = pytesseract.image_to_string(gray)
15        print(text)
16    if key == ord('q'):
17        break
18 cam.release()
19 cv2.destroyAllWindows()
```

##### 4.2 คลิกปุ่ม Run ก็ต้องจะเปิดขึ้นมา นำกระดาษที่มีข้อความภาษาอังกฤษส่งหน้ากล้อง



4.3 ถ้าคลิกปุ่ม s บนคีย์บอร์ดจะแปลงข้อความขึ้นมาในช่อง Shell ถ้าคลิกปุ่ม q จะปิดหน้าต่างลง



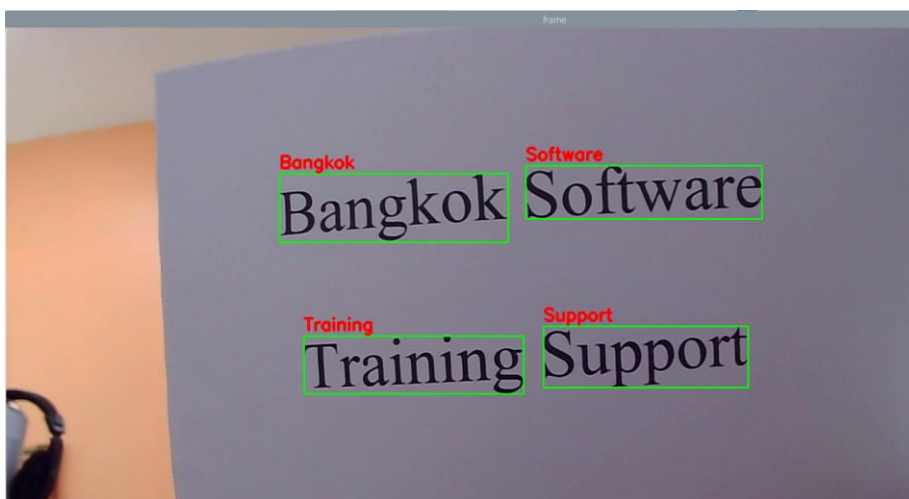
## 5. Workshop 1.3 โปรแกรมแปลงข้อความจากรูปภาพที่ส่งผ่านกล้องเว็บแคมให้เป็นข้อความ แบบที่ 2

### 5.1 เขียนโค้ดคำสั่ง ดังรูป

```

text_detection.py %
1 import cv2
2 import pytesseract
3 from pytesseract import Output
4 cap = cv2.VideoCapture(0)
5 cap.set(cv2.CAP_PROP_BUFFERSIZE, 1)
6 while True:
7     # Capture frame-by-frame
8     ret, frame = cap.read()
9     d = pytesseract.image_to_data(frame, output_type=Output.DICT)
10    n_boxes = len(d['text'])
11    for i in range(n_boxes):
12        if int(d['conf'][i]) > 60:
13            (text, x, y, w, h) = (d['text'][i], d['left'][i], d['top'][i], d['width'][i], d['height'][i])
14            # don't show empty text
15            if text and text.strip() != "":
16                frame = cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)
17                frame = cv2.putText(frame, text, (x, y - 10), cv2.FONT_HERSHEY_SIMPLEX, 1.0, (0, 0, 255), 3)
18            cv2.imshow('frame', frame)
19            if cv2.waitKey(1) & 0xFF == ord('q'):
20                break
21    cap.release() #When everything done, release the capture
22    cv2.destroyAllWindows()
  
```

5.2 คลิกปุ่ม Run กล้องจะเปิดขึ้นมา นำกระดาษที่มีข้อความภาษาอังกฤษส่งหน้ากล้อง



## Workshop 2 Object Detection ตรวจจับวัตถุโดยใช้ YOLOv5

### 1. การทำ Object Detection จากโมเดลตัวอย่าง

1.1 ดาวน์โหลดโมเดล yolov5 โดยเปิด Terminal แล้วพิมพ์คำสั่ง `git clone`

<https://github.com/ultralytics/yolov5.git>

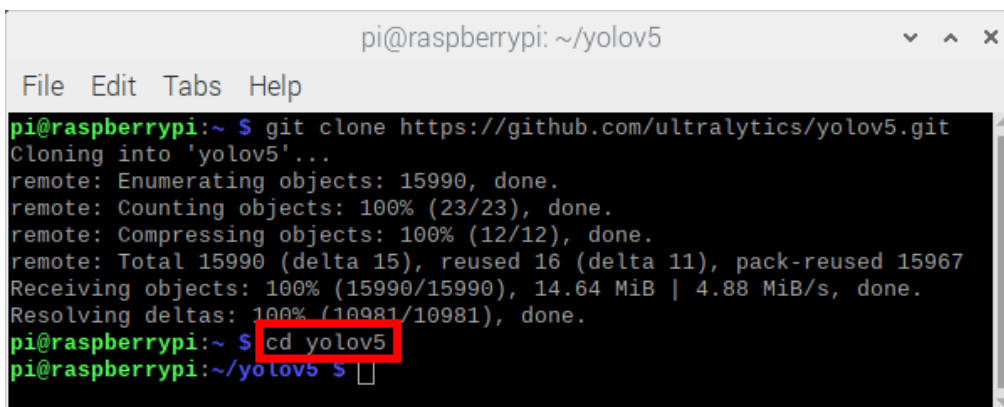


```

pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ git clone https://github.com/ultralytics/yolov5.git

```

1.2 เข้าไปในโฟลเดอร์ yolov5 โดยพิมพ์คำสั่ง `cd yolov5`

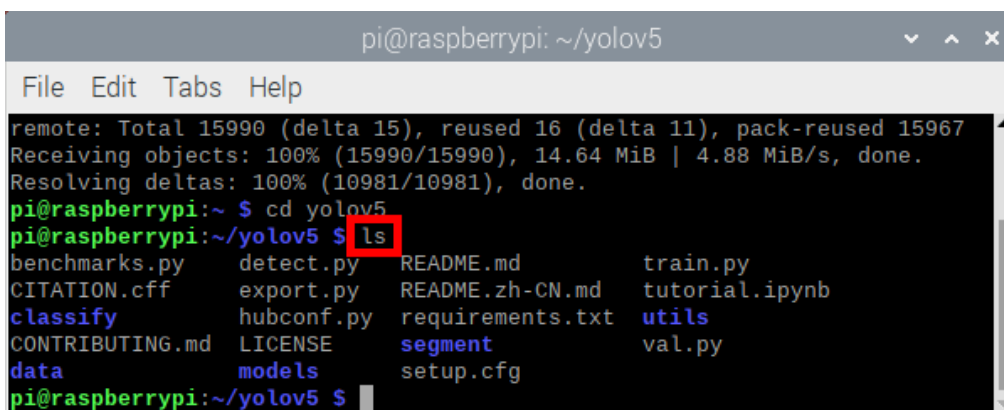


```

pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ git clone https://github.com/ultralytics/yolov5.git
Cloning into 'yolov5'...
remote: Enumerating objects: 15990, done.
remote: Counting objects: 100% (23/23), done.
remote: Compressing objects: 100% (12/12), done.
remote: Total 15990 (delta 15), reused 16 (delta 11), pack-reused 15967
Receiving objects: 100% (15990/15990), 14.64 MiB | 4.88 MiB/s, done.
Resolving deltas: 100% (10981/10981), done.
pi@raspberrypi:~ $ cd yolov5
pi@raspberrypi:~/yolov5 $

```

1.3 พิมพ์คำสั่ง `ls` เพื่อเรียกดูไฟล์ทั้งหมดในโฟลเดอร์ yolov5



```

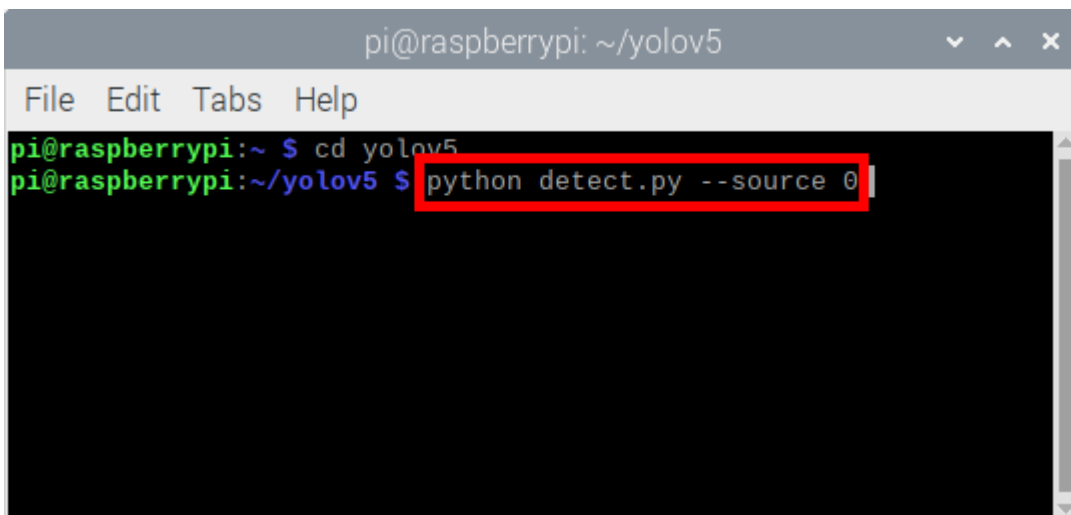
pi@raspberrypi: ~
File Edit Tabs Help
remote: Total 15990 (delta 15), reused 16 (delta 11), pack-reused 15967
Receiving objects: 100% (15990/15990), 14.64 MiB | 4.88 MiB/s, done.
Resolving deltas: 100% (10981/10981), done.
pi@raspberrypi:~ $ cd yolov5
pi@raspberrypi:~/yolov5 $ ls
benchmarks.py  detect.py  README.md  train.py
CITATION.cff  export.py  README.zh-CN.md  tutorial.ipynb
classify       hubconf.py  requirements.txt  utils
CONTRIBUTING.md  LICENSE  segment  val.py
data           models     setup.cfg
pi@raspberrypi:~/yolov5 $

```



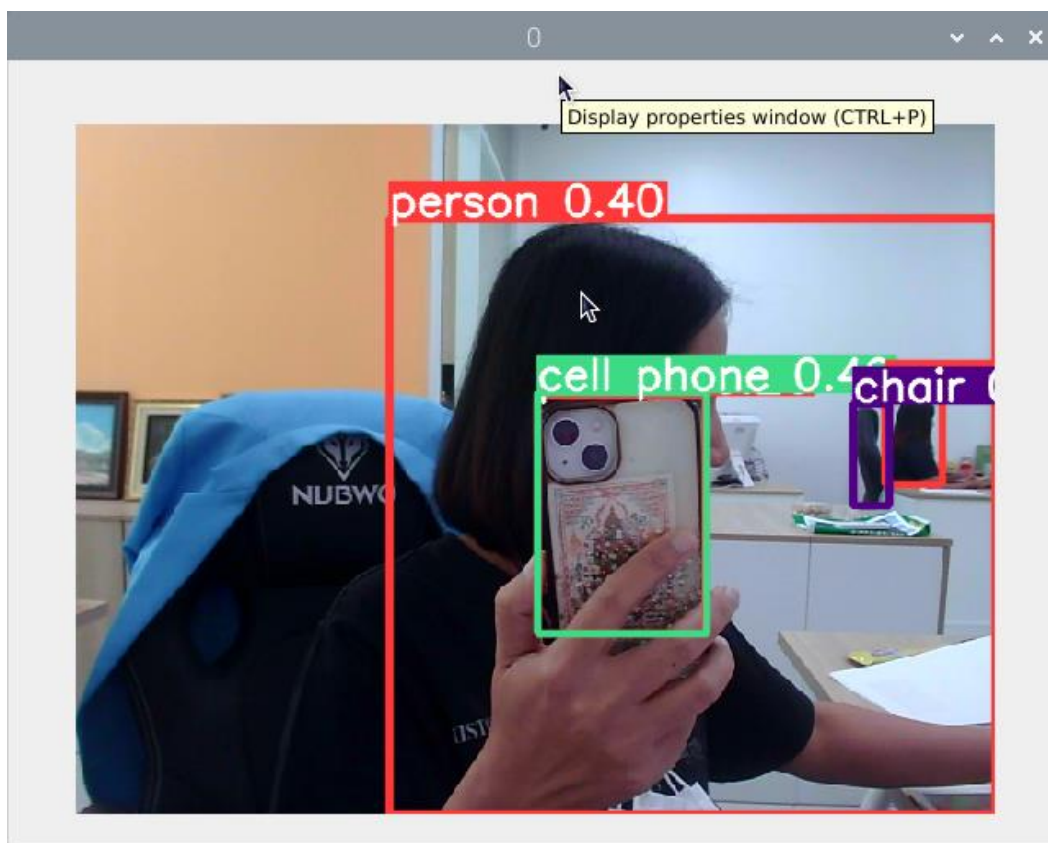


1.7 รันโค้ดคำสั่งตรวจจับวัตถุโดยใช้กล้อง โดยพิมพ์คำสั่ง `python detect.py --source 0`



```
pi@raspberrypi: ~/yolov5
File Edit Tabs Help
pi@raspberrypi:~ $ cd yolov5
pi@raspberrypi:~/yolov5 $ python detect.py --source 0
```

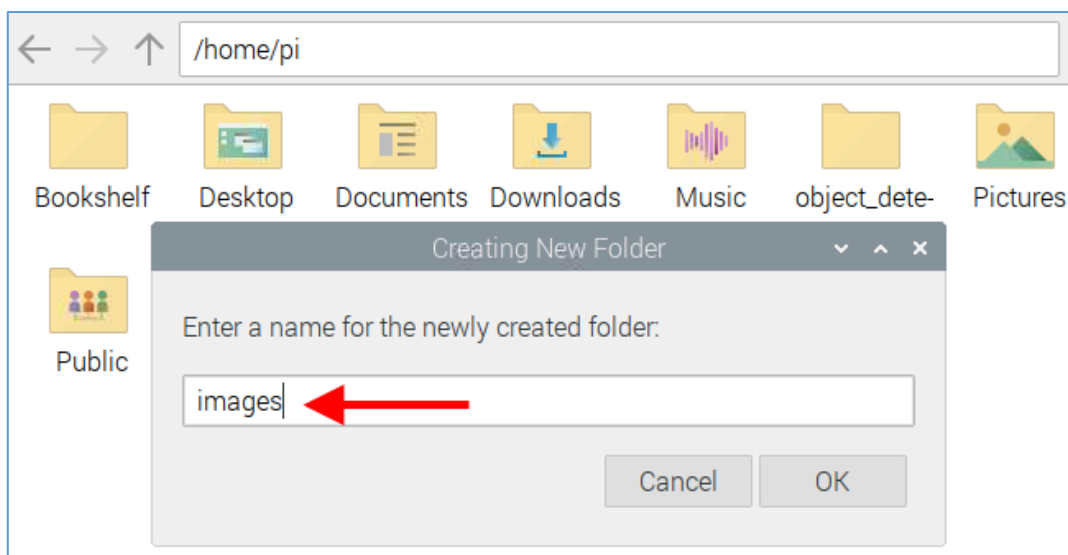
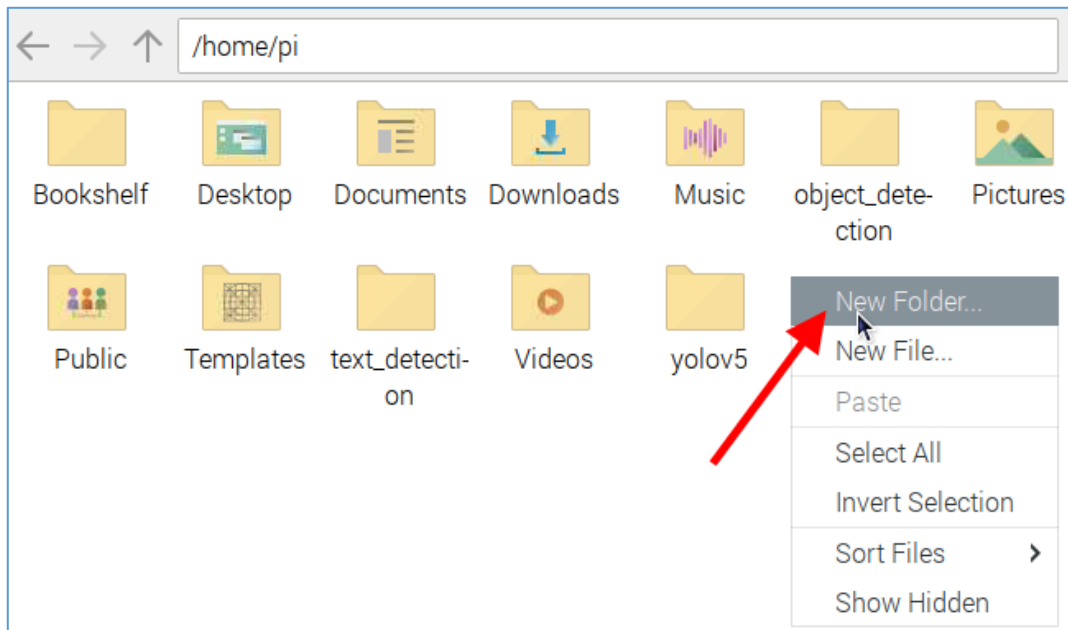
จะได้ผลลัพธ์ ตัวอย่างดังรูป



## 2. การทำ Custom Object Detection ตรวจสอบวัตถุด้วยการสร้างโมเดลของตนเอง

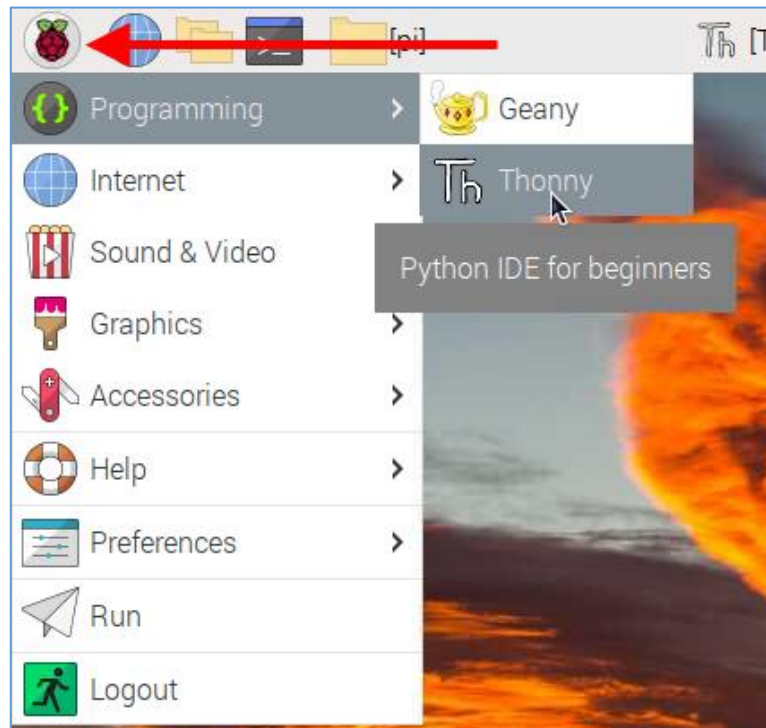
### ขั้นตอนที่ 1 ถ่ายรูปวัตถุที่ต้องการนำมาใช้ในการตรวจจับวัตถุ

1.1 สร้างโฟลเดอร์ images สำหรับเก็บรูปภาพที่ถ่ายไว้ใน /home/pi





## 1.2 เปิดโปรแกรม Thonny

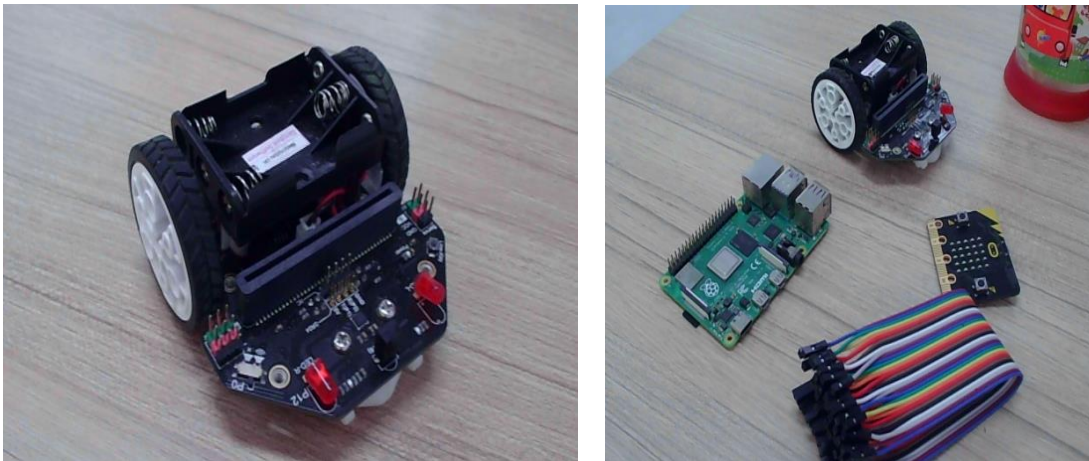


## 1.3 พิมพ์โค้ดคำสั่งสำหรับถ่ายรูป ให้ได้รูปภาพจำนวน 50 รูป (ยิ่งเยอะยิ่งดี) จากนั้นคลิกปุ่ม Run

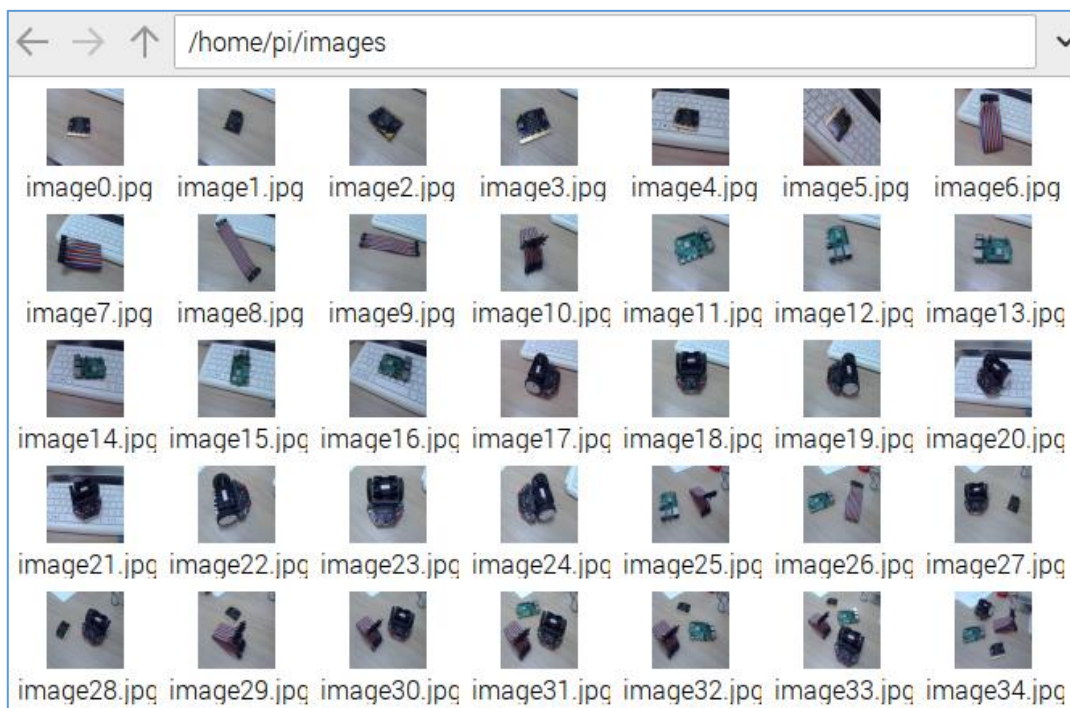
```
capture1.py x
1 import cv2
2 import time
3
4 i = 0
5
6 cap = cv2.VideoCapture(0)
7
8 while True:
9     img, frame = cap.read()
10    frame = cv2.resize(frame, (640, 640))
11    cv2.imshow("Capture", frame)
12
13    if cv2.waitKey(1) & 0xFF == ord('s'):
14        cv2.imwrite("/home/pi/images/image%d.jpg" %i, frame)
15        i += 1
16    if cv2.waitKey(1) & 0xFF == 27:
17        break
18 cap.release()
19 cv2.destroyAllWindows()
```



1.4 กล้องจะเปิดขึ้นมา ให้นำกล้องมาส่องถ่ายภาพวัตถุ โดยกดปุ่ม s บนคีย์บอร์ดเพื่อถ่ายแต่ละรูป ซึ่งจะต้องปรับเปลี่ยนรูปภาพให้หลากหลายมุมมอง โดยถ่ายทั้งแบบเดี่ยวและแบบกลุ่ม ตัวอย่างเช่น

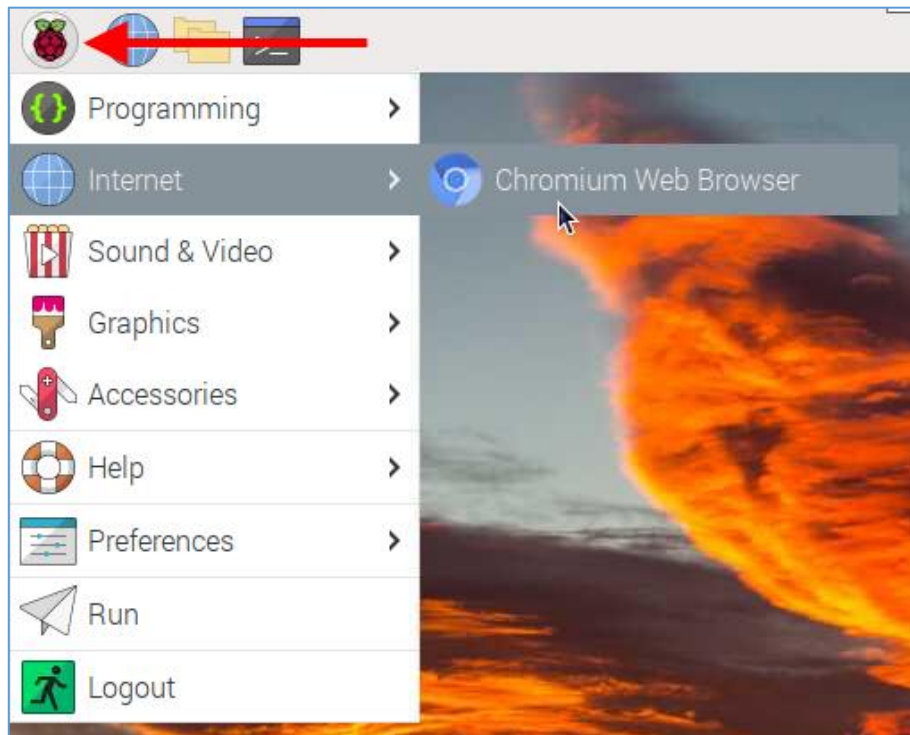


1.5 เปิดโฟลเดอร์ /home/pi/images หากมีรูปภาพใดไม่ต้องการก็สามารถลบทิ้งได้

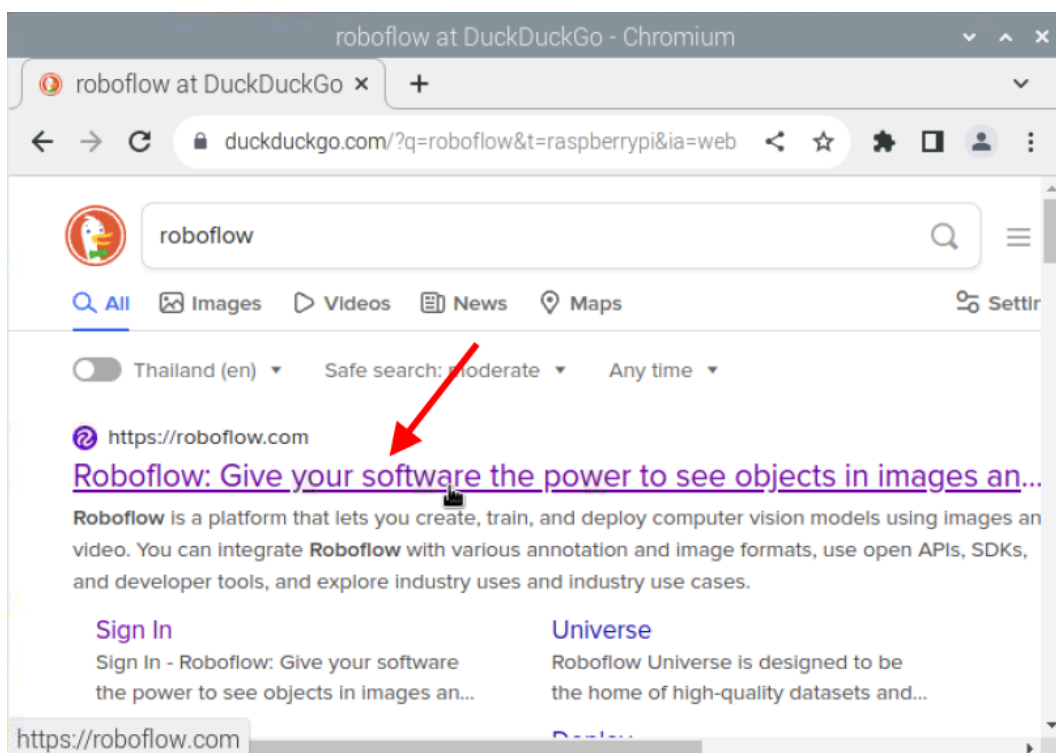


## ขั้นตอนที่ 2 ติดป้ายกำกับรูปภาพด้วย RoboFlow

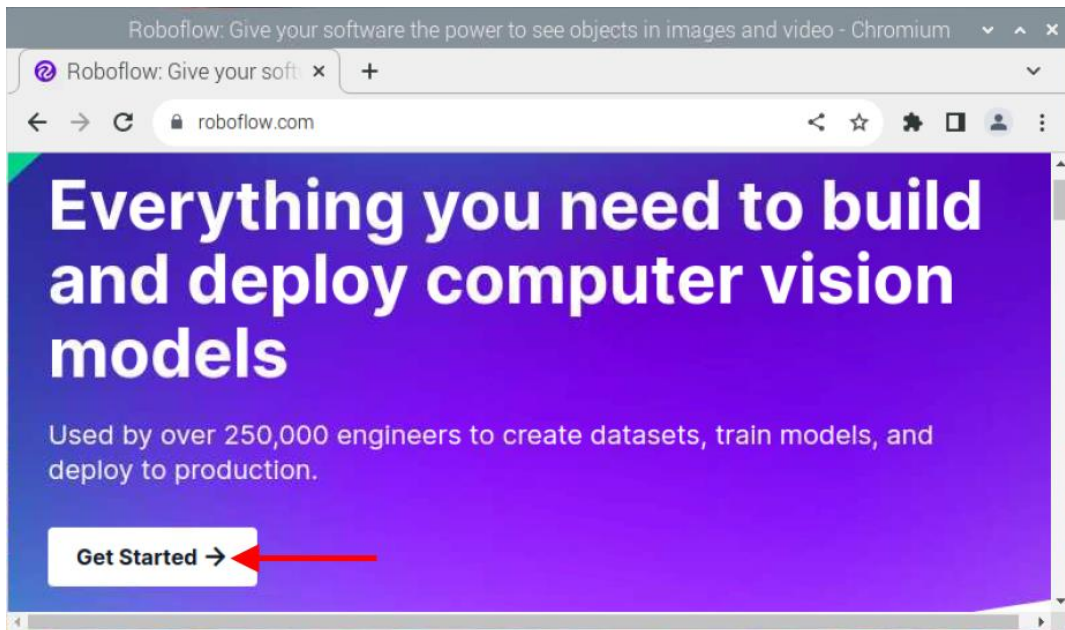
### 2.1 เปิดโปรแกรม Chromium Web Browser



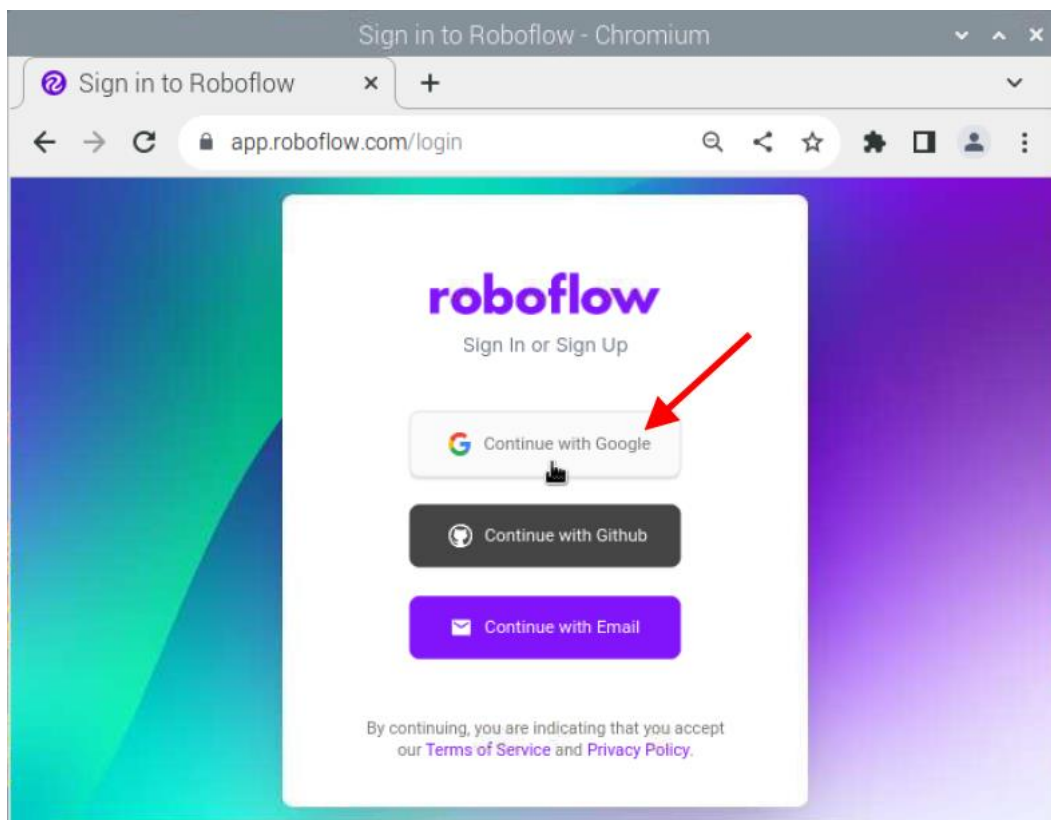
### 2.2 พิมพ์คำค้นหาว่า “roboflow” จากนั้นคลิกเข้าเว็บไซต์ <https://roboflow.com>



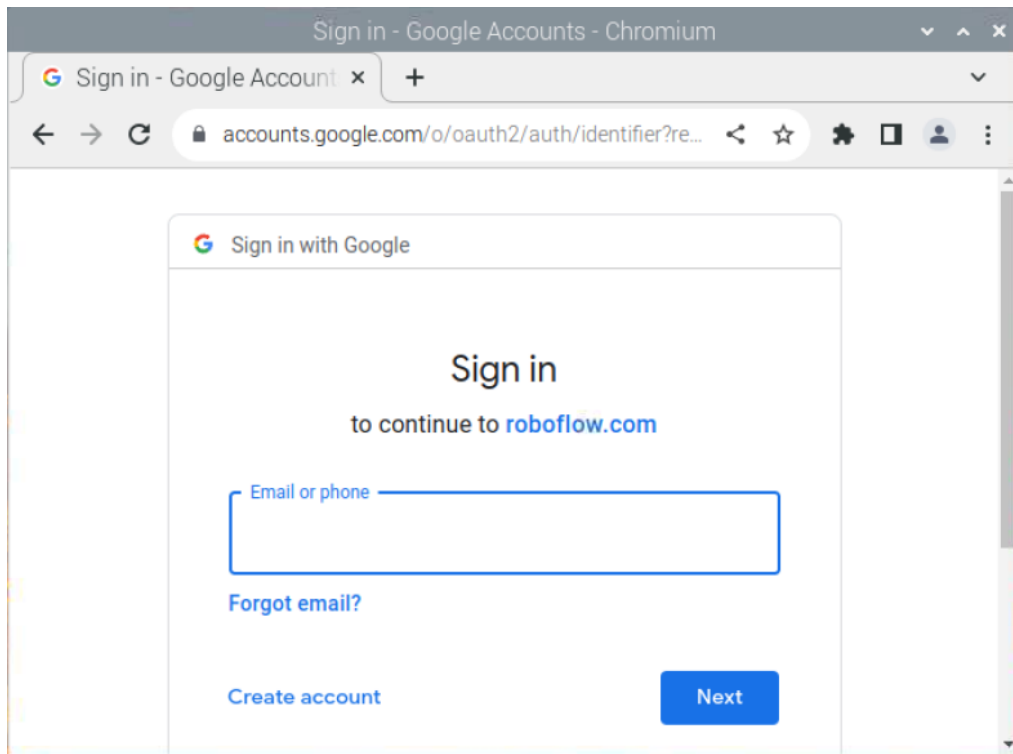
### 2.3 คลิกปุ่ม Get Started -->



### 2.4 เลือกล็อกอินด้วยบัญชี Gmail โดยคลิกปุ่ม Continue with Google

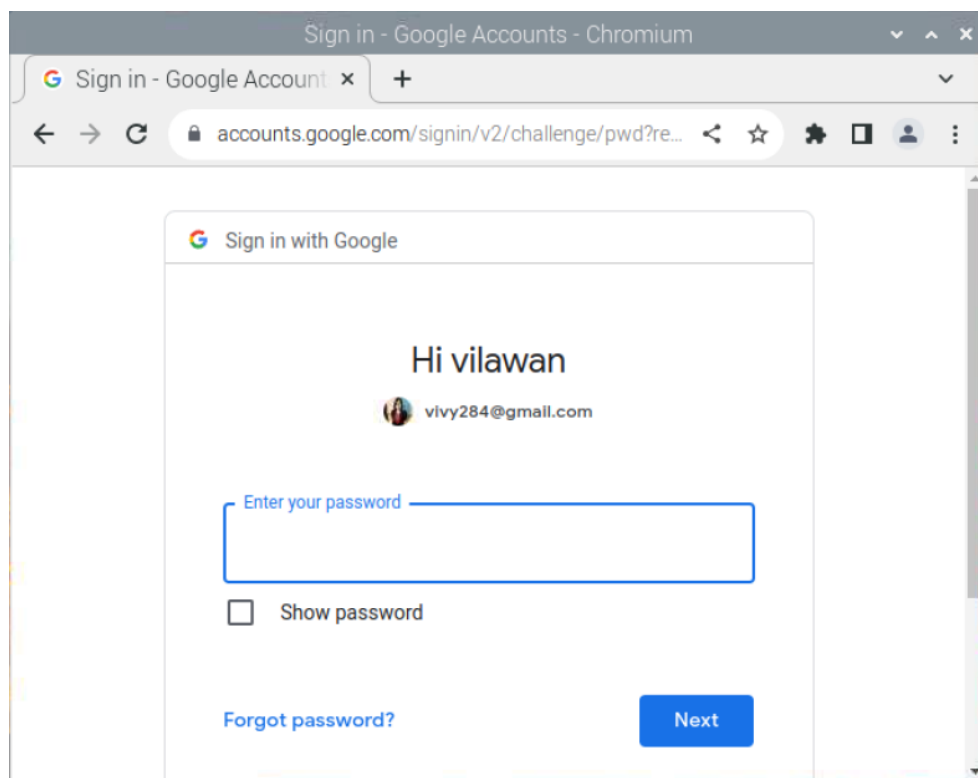


## 2.5 กรอกอีเมล แล้วคลิกปุ่ม Next



The screenshot shows a web browser window titled "Sign in - Google Accounts - Chromium". The address bar displays "accounts.google.com/o/oauth2/auth/identifier?re...". The main content area features a "Sign in with Google" header and a "Sign in" section. Below the header, it says "Sign in to continue to roboflow.com". There is a text input field labeled "Email or phone" with a blue border. Below the input field are links for "Forgot email?" and "Create account". A blue "Next" button is positioned at the bottom right of the sign-in area.

## 2.6 กรอกรหัสผ่าน แล้วคลิกปุ่ม Next



The screenshot shows the same web browser window, but the address bar now displays "accounts.google.com/signin/v2/challenge/pwd?re...". The main content area shows a "Sign in with Google" header and a personalized greeting: "Hi vilawan" with a profile picture and the email address "vivvy284@gmail.com". Below the greeting is a text input field labeled "Enter your password" with a blue border. Underneath the input field is a checkbox labeled "Show password". At the bottom left, there is a link for "Forgot password?". A blue "Next" button is located at the bottom right of the sign-in area.

## 2.7 เลือกวัตถุประสงค์การใช้งาน จากนั้นคลิกปุ่ม Continue

**Let's set up your new workspace!**

A workspace is where a team can collaborate to create, manage, and label datasets, and train and deploy models.

● ● ●

To start, let's customize it.

What will you be using this workspace for?

Workspace Name: E.g. full name or username.

## 2.8 คลิก Skip เพื่อข้ามขั้นตอนการเชิญผู้อื่นมาร่วมใช้งาน

**Let's set up your new workspace!**

A workspace is where a team can collaborate to create, manage, and label datasets, and train and deploy models.

● ● ●

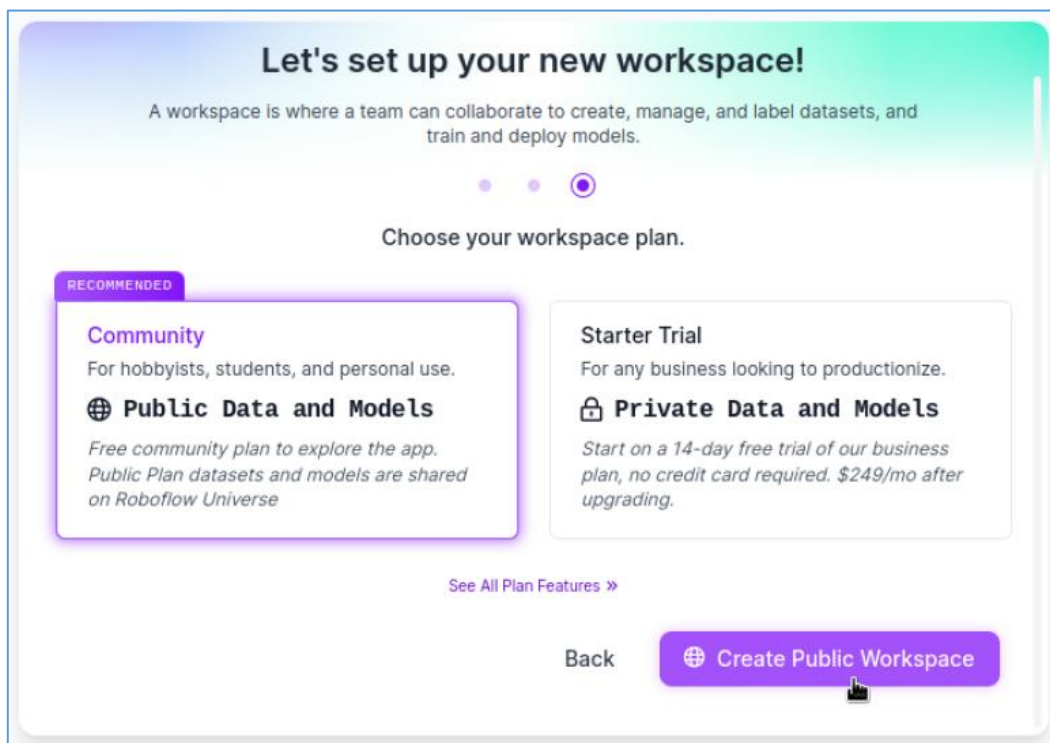
Invite others to collaborate on projects.

Invite via email 2 invites available

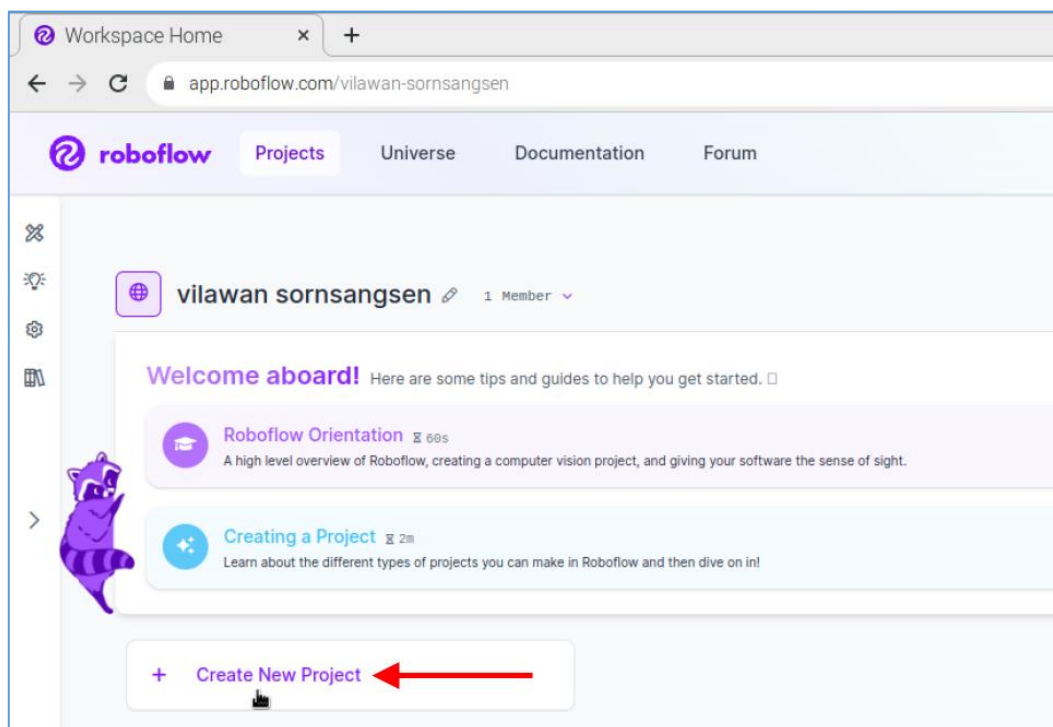
Role:



2.9 คลิกเลือก Public Data and Models จากนั้นคลิกปุ่ม Create Public Workspace



2.10 คลิกสร้างโปรเจกต์ใหม่ โดยคลิกที่ Create New Project




2.11 เลือก Project Type เป็นแบบ Object Detection ตั้งชื่อ Project Name และระบุต้องการตรวจจับ  
สิ่งใด จากนั้นคลิกปุ่ม Create Public Project


**Create New Project** [X]

vilawan sornsangsen / 🌐 New Public Project


**Project Type** Please select a type



**Classification**  
Assign labels to the entire image.



**Object Detection**  
Find multiple things and their specific location.



**Instance Segmentation**  
Detect multiple objects and their actual shape.

Show More ↓

**Project Name**

coding\_object

**What are you detecting?** ⓘ

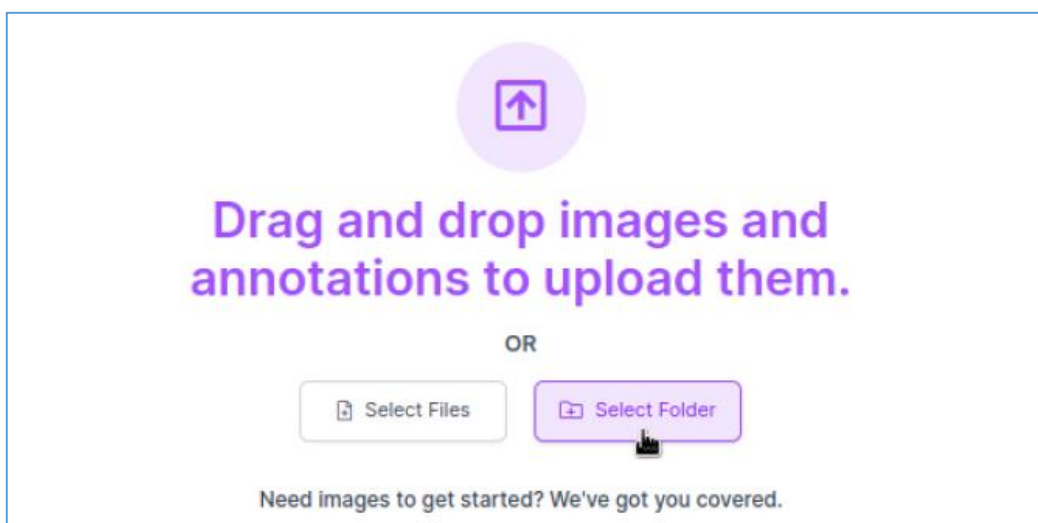
coding board

**License**

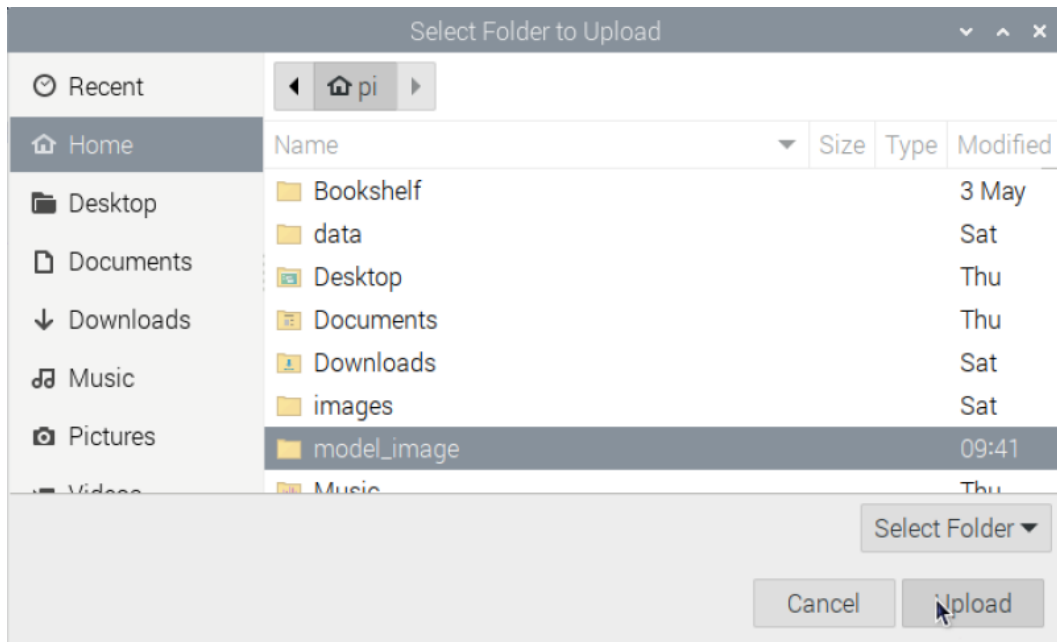
CC BY 4.0

Cancel Create Public Project

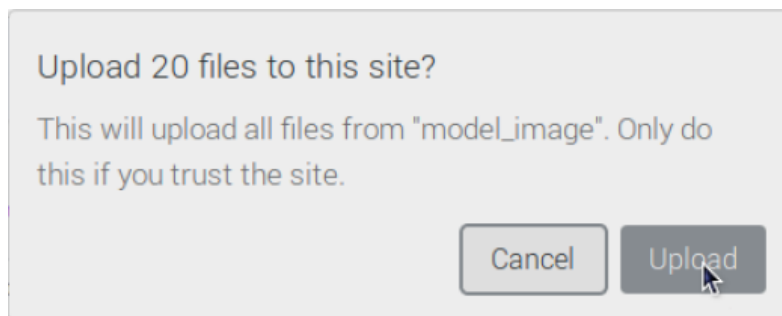
2.12 อัปโหลดไฟล์เดือรูปภาพ โดยคลิก Select Folder



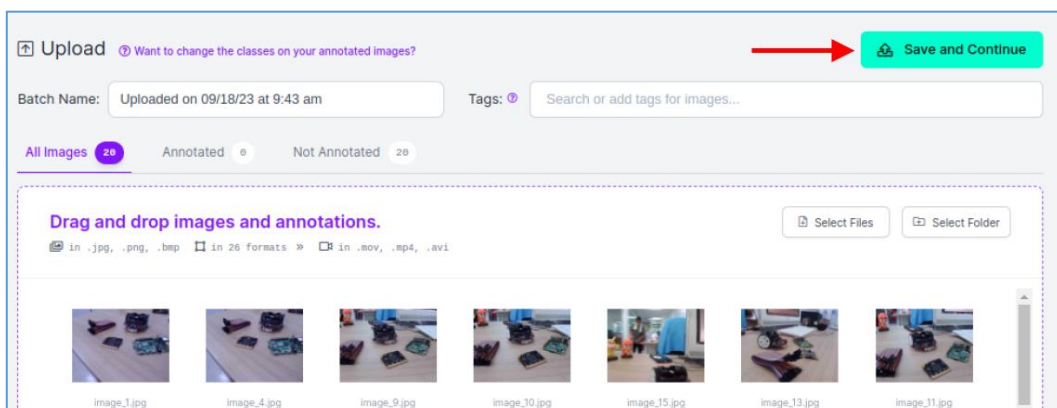
### 2.13 เลือกโฟลเดอร์ที่เก็บรูปภาพ จากนั้นคลิกปุ่ม Upload



### 2.14 คลิกปุ่ม Upload

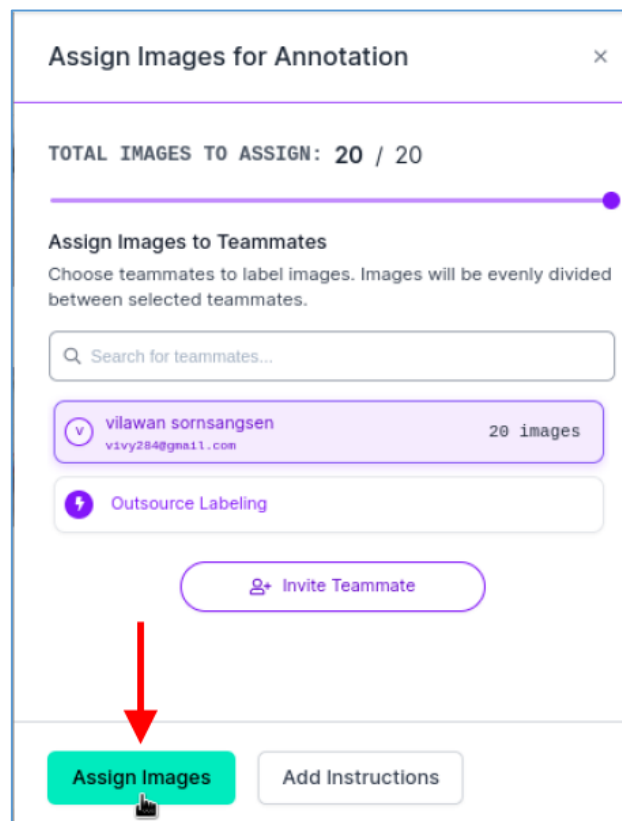


### 2.15 คลิกปุ่ม Save and Continue

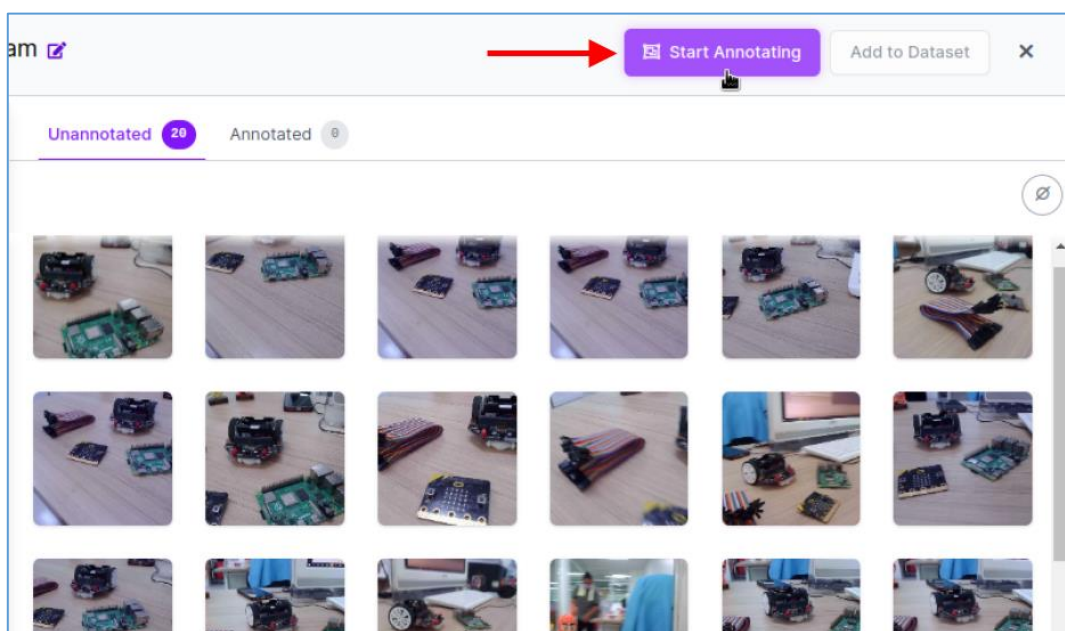




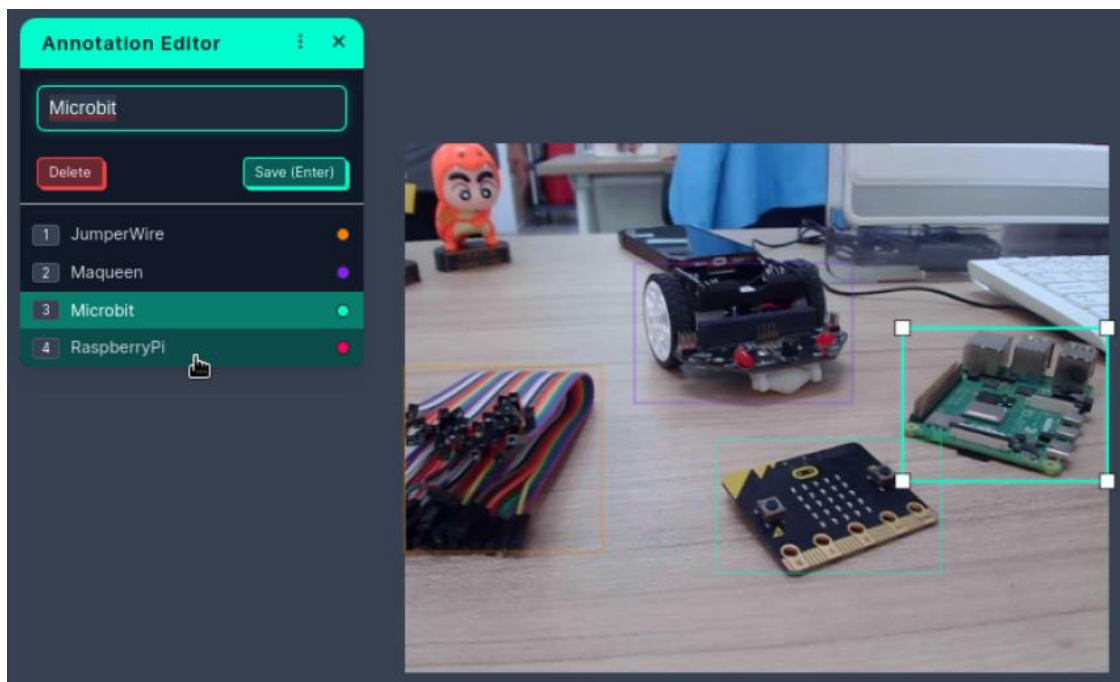
## 2.16 คลิกปุ่ม Assign Images



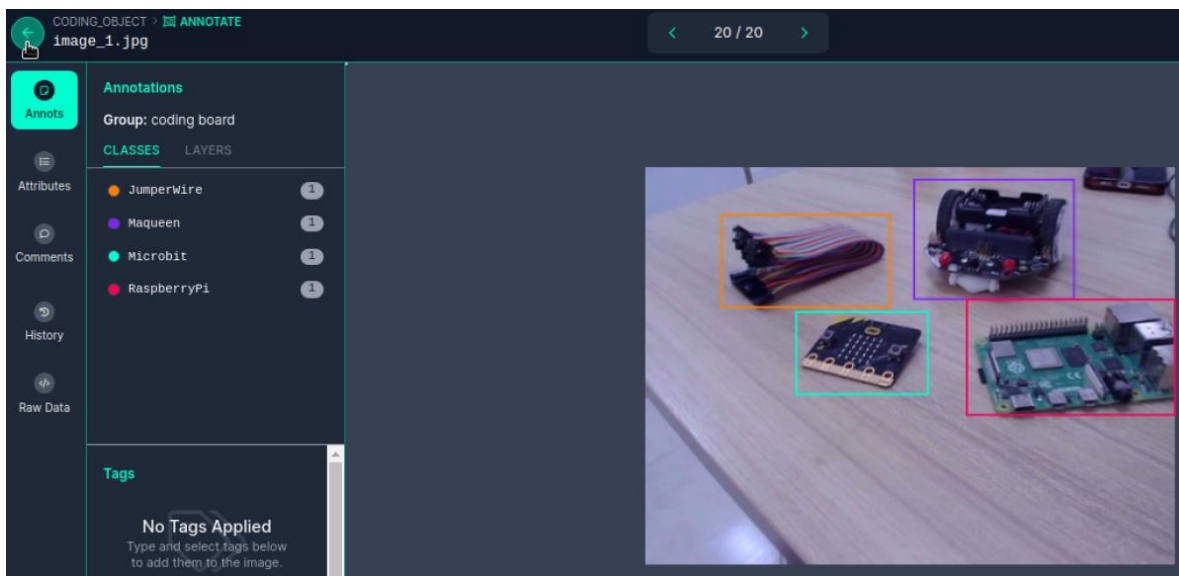
## 2.17 เริ่มต้นการสร้างป้ายกำกับ โดยคลิกปุ่ม Start Annotating



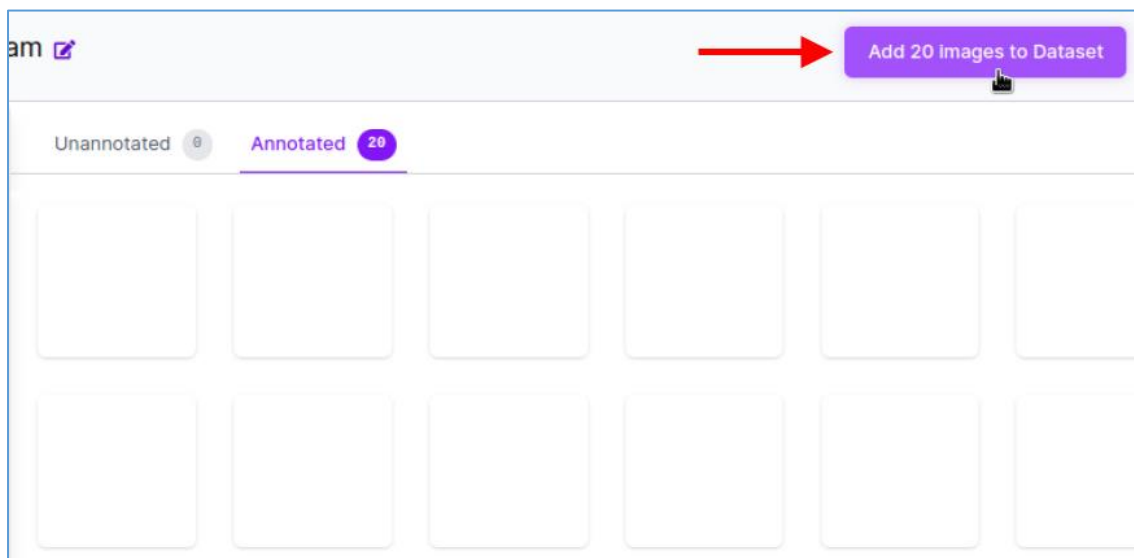
## 2.18 วาดกรอบล้อมรอบวัตถุ จากนั้นใส่ป้ายกำกับวัตถุลงไป



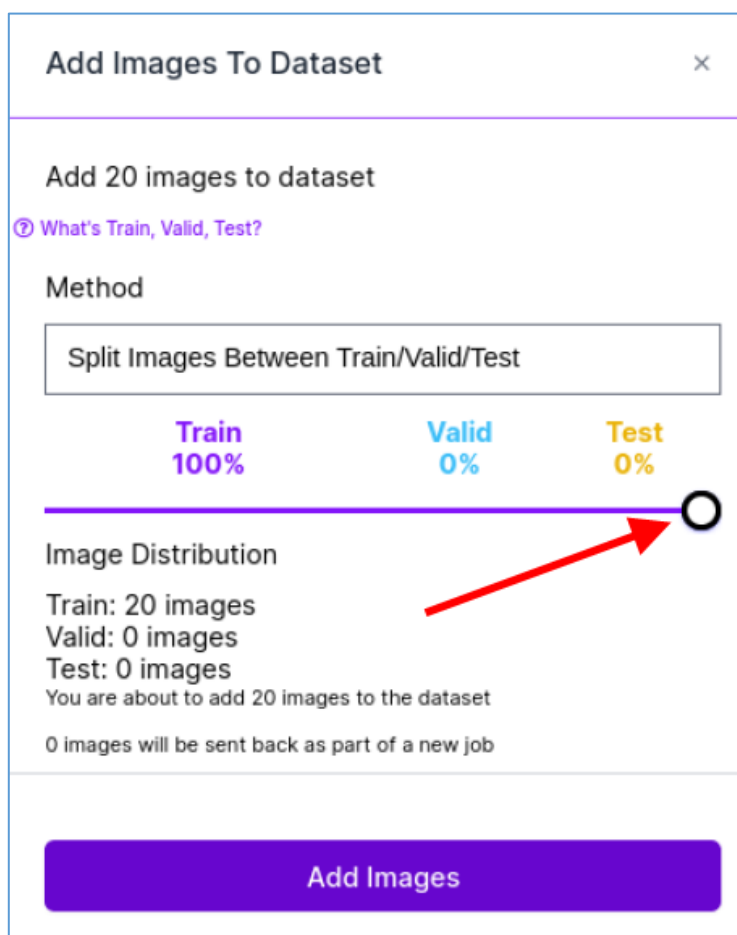
## 2.19 คลิกปุ่มลูกศร > เพื่อเลื่อนรูป สร้างป้ายกำกับให้ครบทุกรูป เสร็จแล้วคลิกปุ่ม <-- ย้อนกลับ



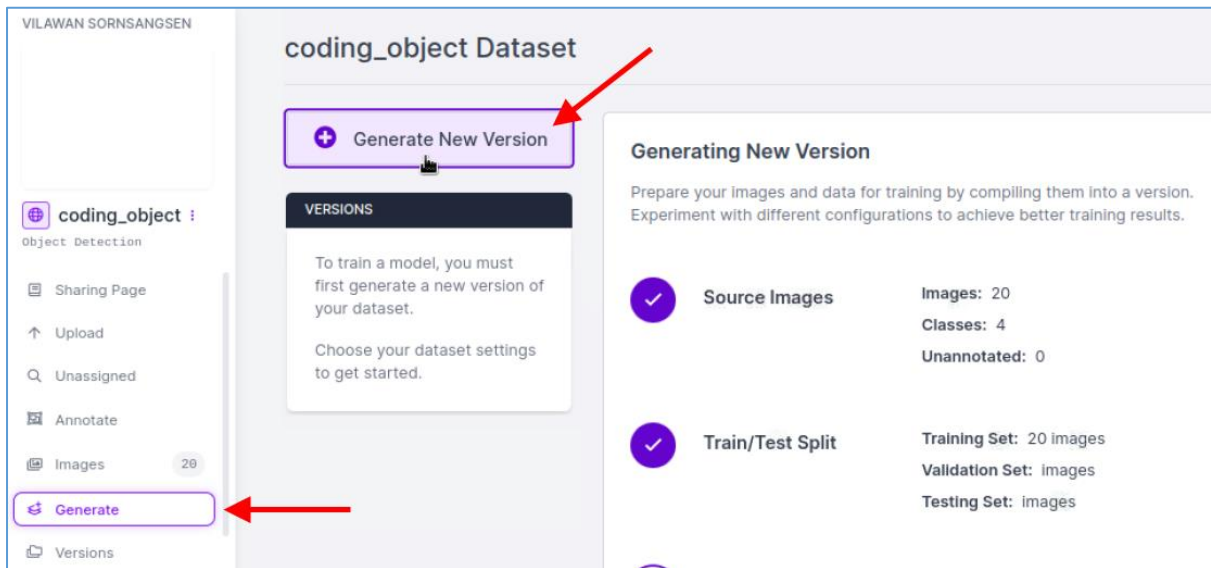
## 2.20 คลิกปุ่ม Add ... images to Dataset



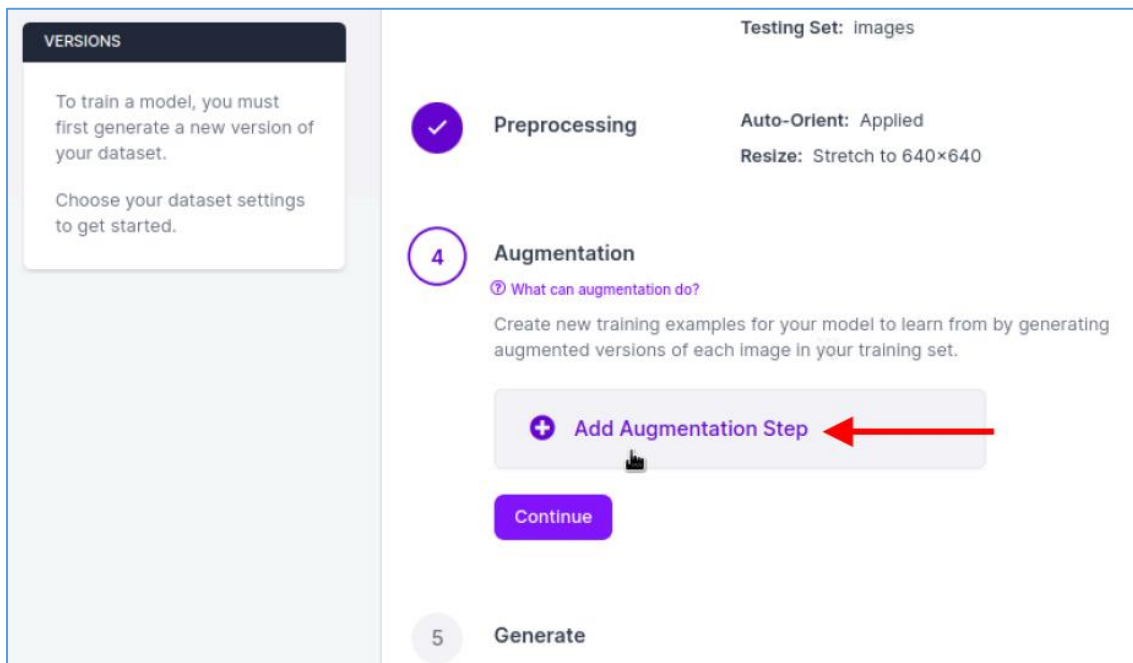
## 2.21 ปรับสัดส่วน Dataset โดยเลือกรูปภาพสำหรับการ Train เป็น 100% แล้วคลิกปุ่ม Add Images



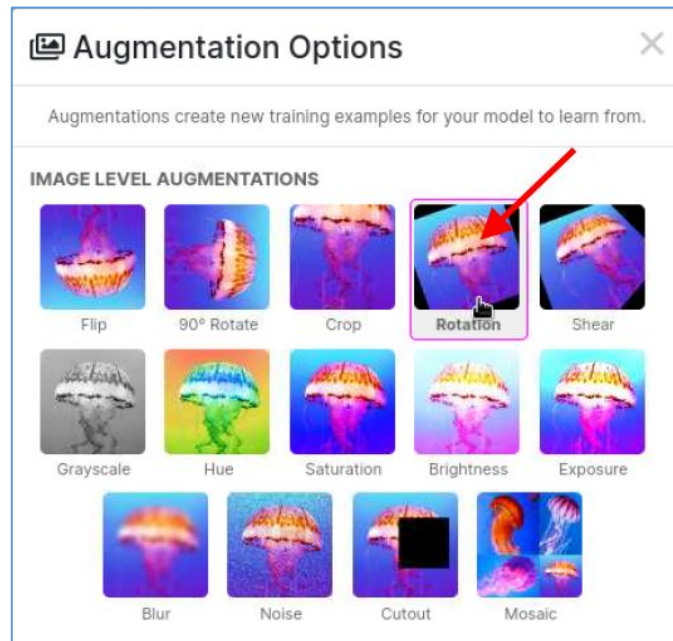
2.22 ขยายรูปภาพสำหรับการ Train ให้มากขึ้นเป็น 3 เท่า โดยคลิกที่ปุ่ม Generate ที่อยู่ด้านซ้ายมือ จากนั้นคลิกปุ่ม Generate New Version



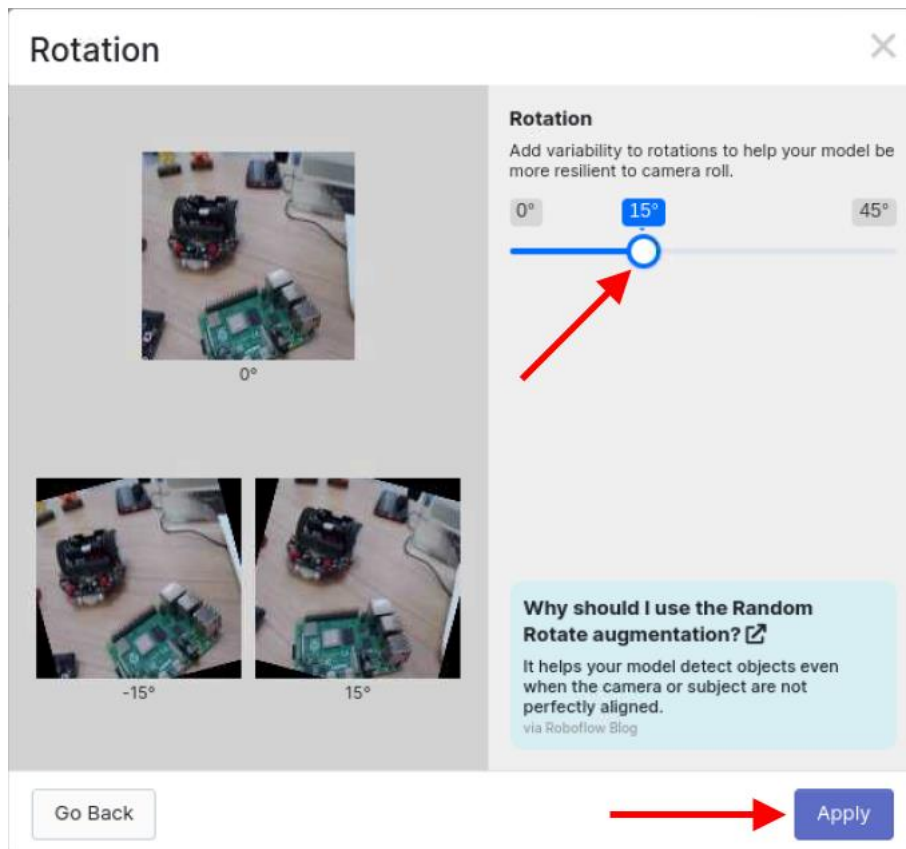
2.23 คลิกเลือกรูปแบบการเปลี่ยนแปลงรูปภาพ โดยคลิกที่ Add Augmentation Step



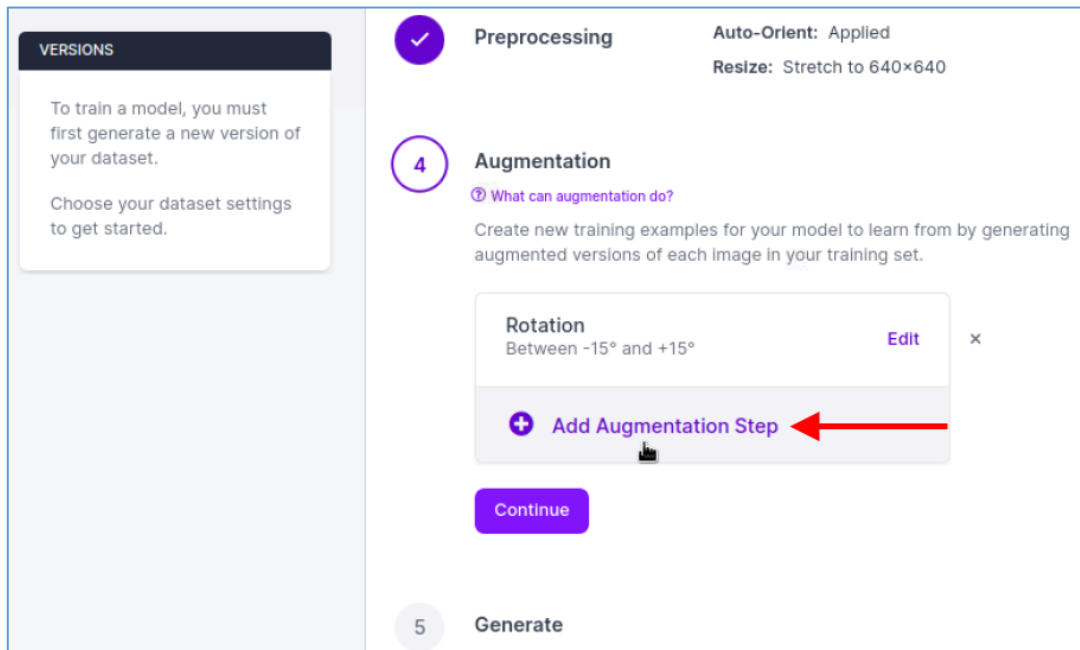
2.24 เลือกรูปแบบที่ต้องการ ในที่นี้จะเลือกการหมุนรูปภาพด้วย Rotation



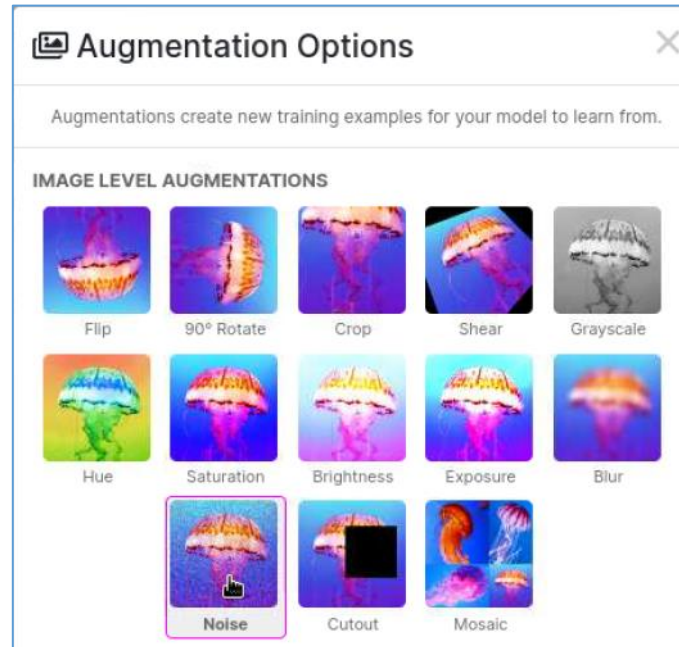
2.25 เลื่อนองศาในการหมุน โดยดูตัวอย่างได้จากรูปภาพด้านซ้ายมือ จากนั้นคลิกปุ่ม Apply



## 2.26 เพิ่มรูปแบบการเปลี่ยนแปลงรูปภาพ โดยคลิกที่ Add Augmentation Step



## 2.27 คลิกเลือกรูปแบบที่ต้องการ ในที่นี้จะเพิ่ม Noise เข้าไป





## 2.28 ปรับค่า Noise จากนั้นคลิกปุ่ม Apply

**Noise**

Add noise to help your model be more resilient to camera artifacts.

0% **5%** 25%

0%

5%

**Why would I add noise to my images?**

Noise can help defend against adversarial attacks and prevent overfitting.  
via Roboflow Blog

Go Back Apply

## 2.29 คลิกปุ่ม Continue

**VERSIONS**

To train a model, you must first generate a new version of your dataset.

Choose your dataset settings to get started.

**Preprocessing** Auto-Orient: Applied  
Resize: Stretch to 640x640

**4 Augmentation**  
[? What can augmentation do?](#)

Create new training examples for your model to learn from by generating augmented versions of each image in your training set.

**Rotation** Between -15° and +15° Edit ×

**Noise** Up to 5% of pixels Edit ×

**+** Add Augmentation Step

Continue

### 2.30 จะเห็นว่าระบบจะสร้างรูปภาพเพิ่มขึ้นมา 3 เท่าจากเดิม จากนั้นคลิกปุ่ม Generate

**VERSIONS**

To train a model, you must first generate a new version of your dataset.

Choose your dataset settings to get started.

- ✓ **Preprocessing**
  - Auto-Orient: Applied
  - Resize: Stretch to 640×640
- ✓ **Augmentation**
  - Rotation: Between -15° and +15°
  - Noise: Up to 5% of pixels
- 5 **Generate**

Review your selections and select a version size to create a moment-in-time snapshot of your dataset with the applied transformations.

Larger versions take longer to train but often result in better model performance. [See how this is calculated »](#)

Maximum Version Size

60 images (3x) ▾

**Generate** ←

### 2.31 ส่งออกรูปภาพ Dataset โดยคลิกปุ่ม Export Dataset

v1 **2023-09-18 10:36am** **Export Dataset** ⋮

Generated on Sep 18, 2023

**This version doesn't have a model.**

Train an optimized, state of the art model with Roboflow or upload a custom trained model to use features like Label Assist and Model Evaluation and deployment options like our auto-scaling API and edge device support.

**Train with Roboflow** Custom Train and Upload

Available Credits: 3

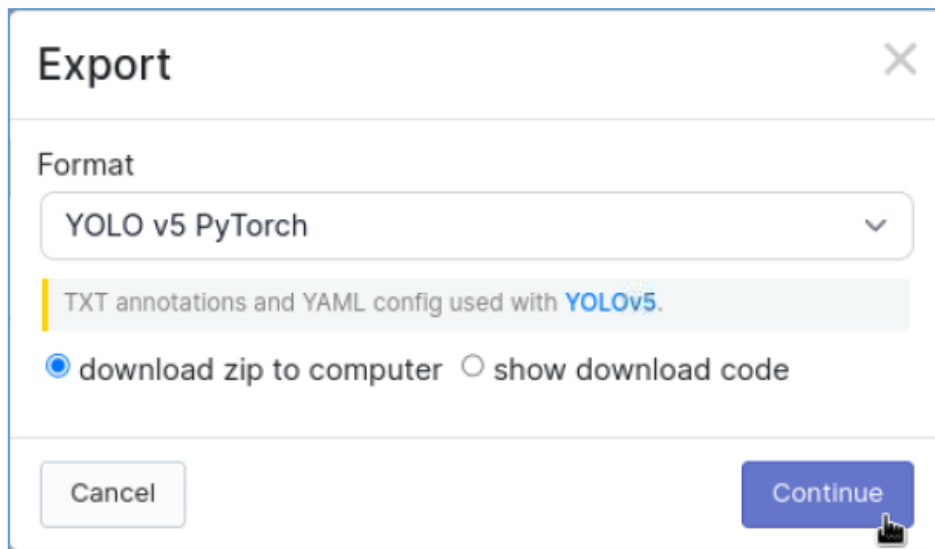


### 2.32 เลือกรูปแบบการดาวน์โหลด ดังนี้

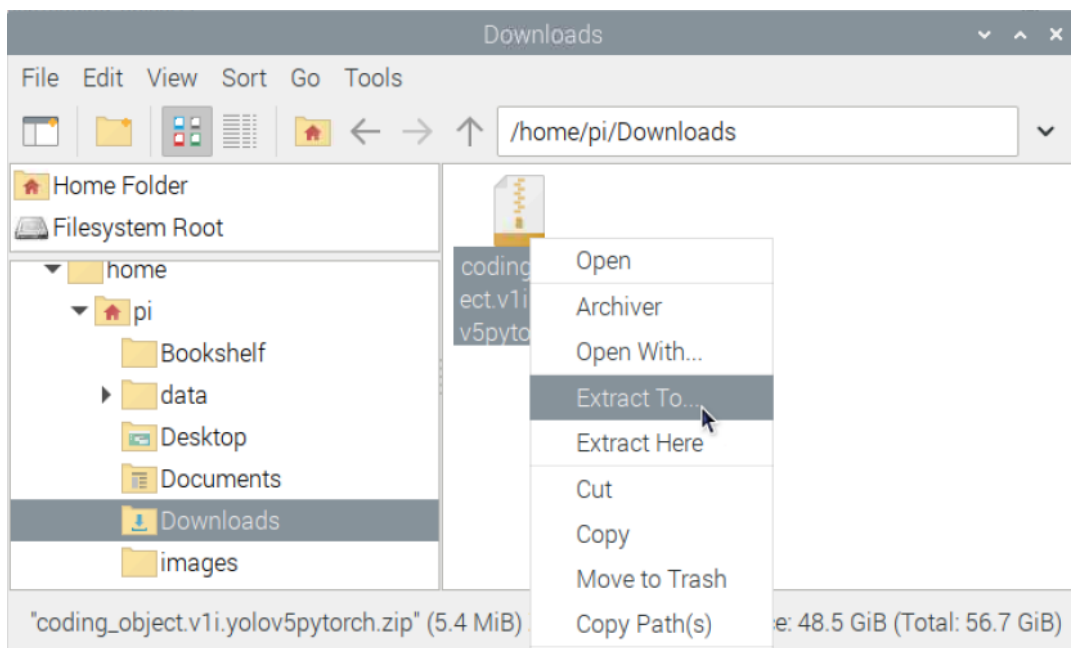
- Format: YOLO v5 PyTorch

- download zip to computer

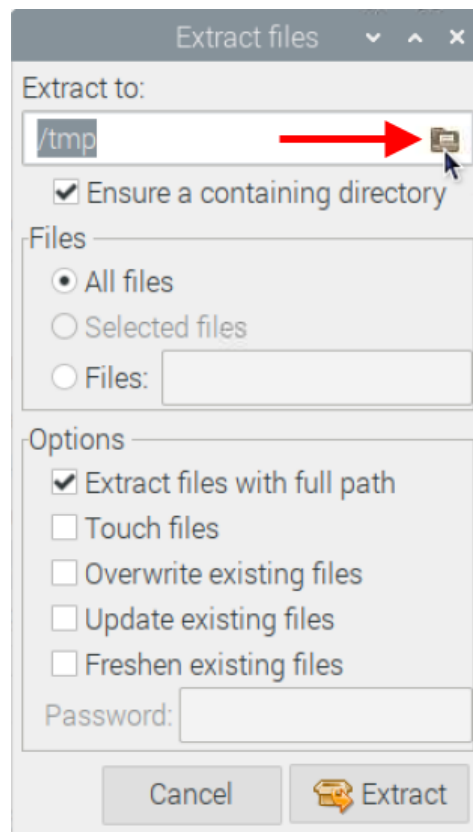
จากนั้นคลิกปุ่ม Continue



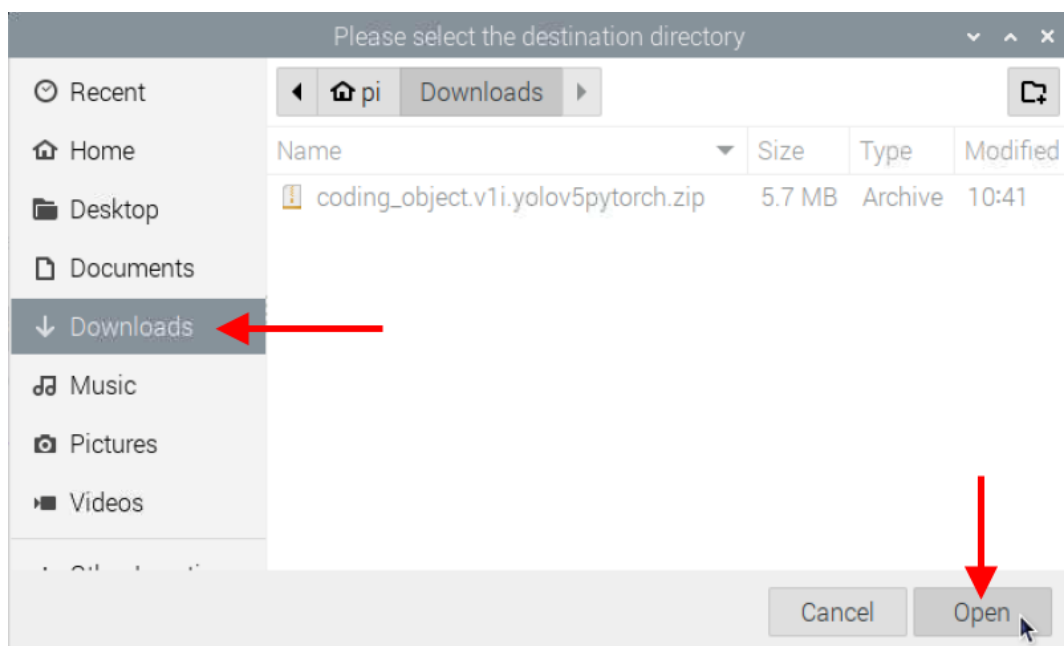
2.33 ไฟล์จะถูกดาวน์โหลดเข้ามาในโฟลเดอร์ /home/pi/Downloads จากนั้นให้คลิกขวาที่ไฟล์เลือกคำสั่ง Extract To....



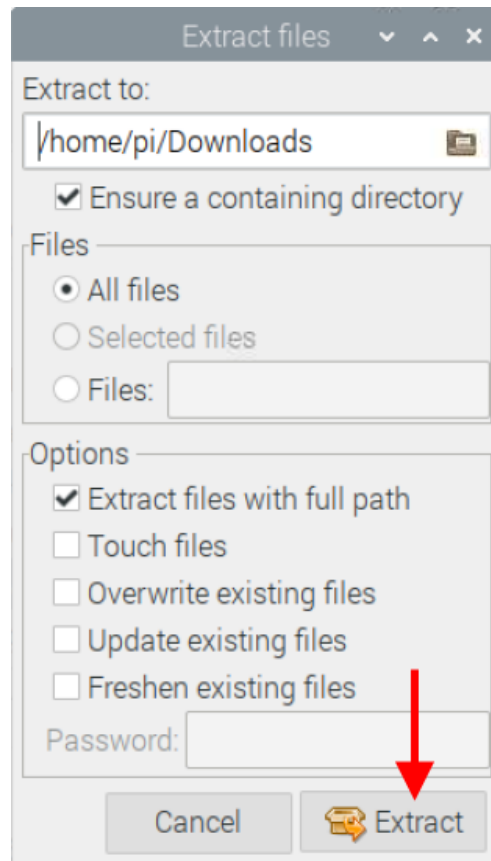
### 2.34 คลิกเลือกโฟลเดอร์ปลายทาง



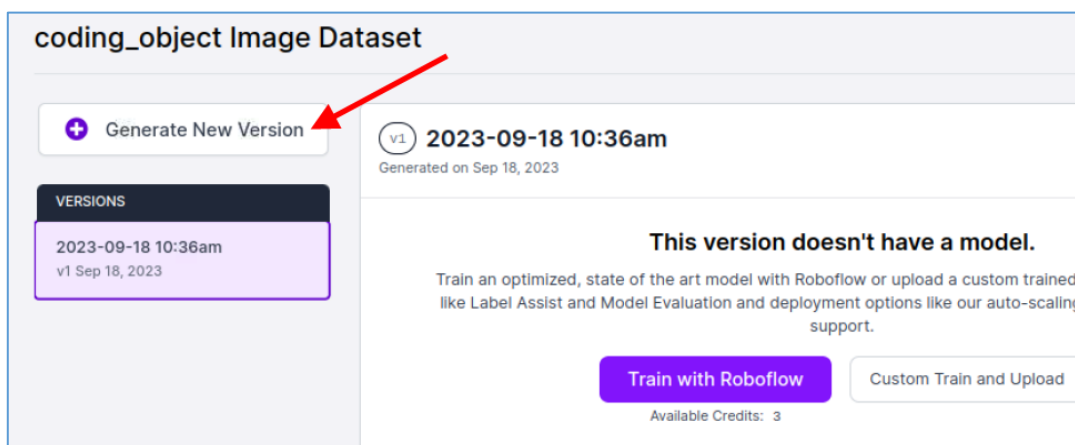
### 2.35 คลิกเลือกโฟลเดอร์ Downloads จากนั้นคลิกปุ่ม Open



### 2.36 คลิกปุ่ม Extract



### 2.37 หากต้องการขยายรูปภาพในการ Train เพิ่ม สามารถคลิกที่ Generate New Version อีกครั้ง



### 2.38 คลิกปุ่ม Continue

**3 Preprocessing**

[What can preprocessing do?](#)

Decrease training time and increase performance by applying image transformations to all images in this dataset.

Auto-Orient	Edit	x
Resize Stretch to 640x640	Edit	x

**+ Add Preprocessing Step**

**Continue**

### 2.39 ลบรูปแบบ Augmentation เดิมทิ้งไป

**4 Augmentation**

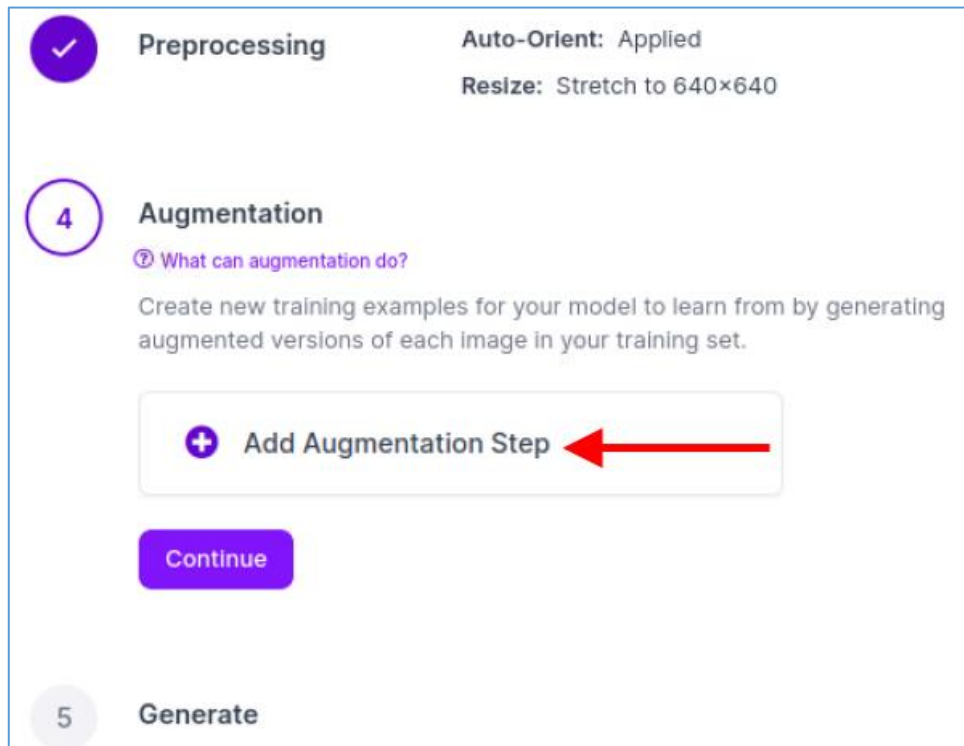
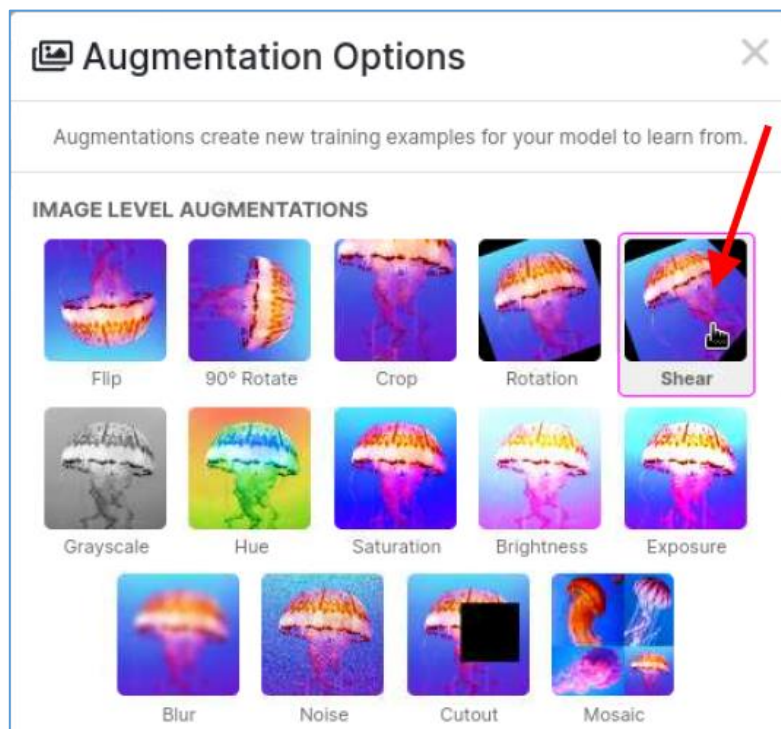
[What can augmentation do?](#)

Create new training examples for your model to learn from by generating augmented versions of each image in your training set.

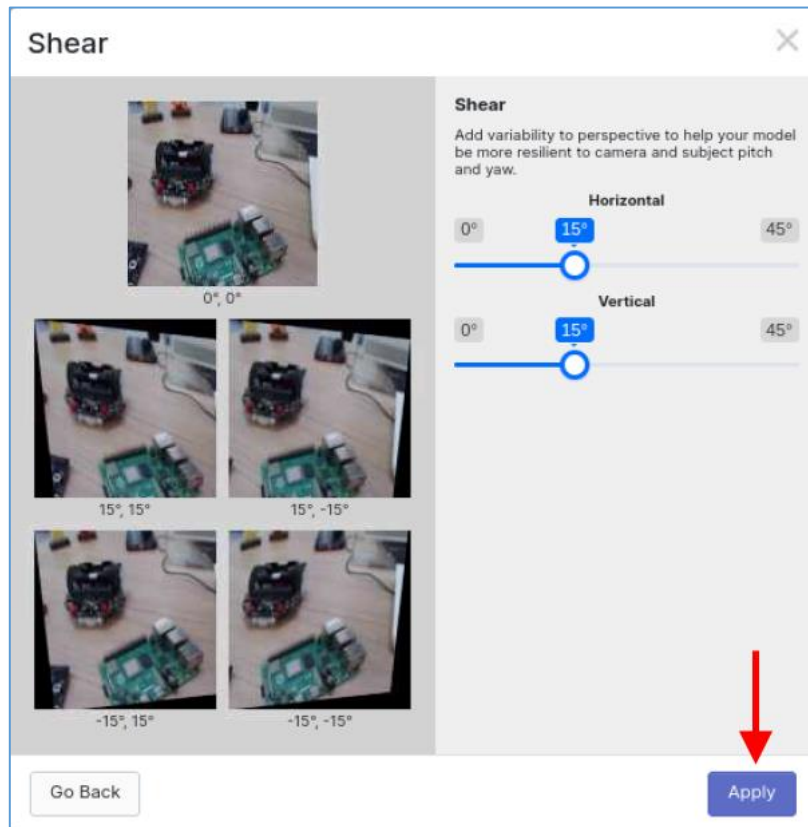
Rotation Between -15° and +15°	Edit	x
Noise Up to 5% of pixels	Edit	x

**+ Add Augmentation Step**

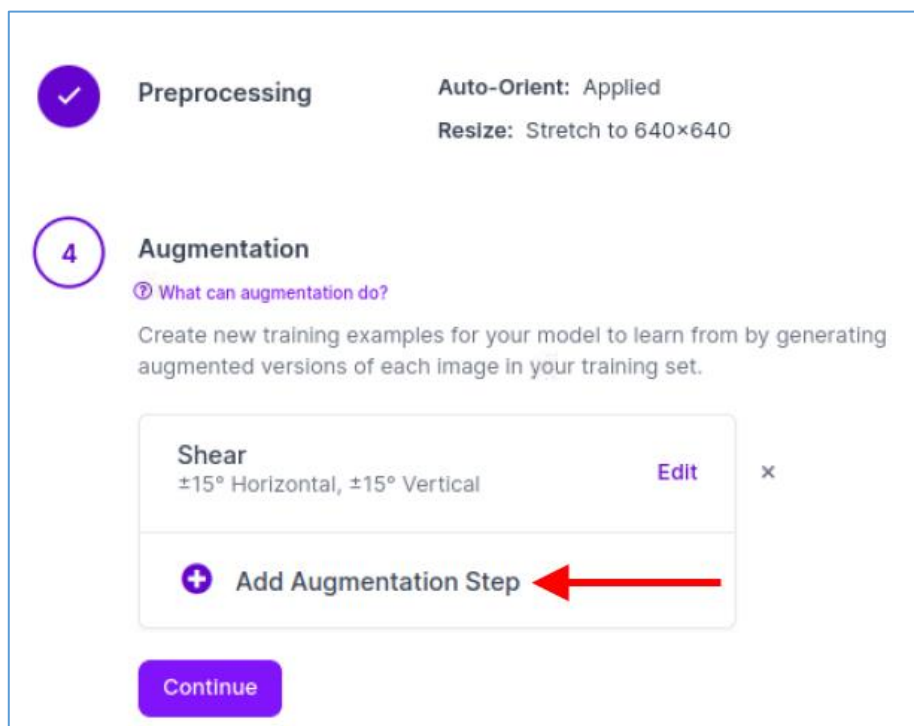
**Continue**

2.40 คลิก Add Augmentation Step2.41 คลิกเลือกรูปแบบการเปลี่ยนแปลงรูปภาพ ในที่นี้เลือกเอียงรูปภาพ โดยเลือก Shear

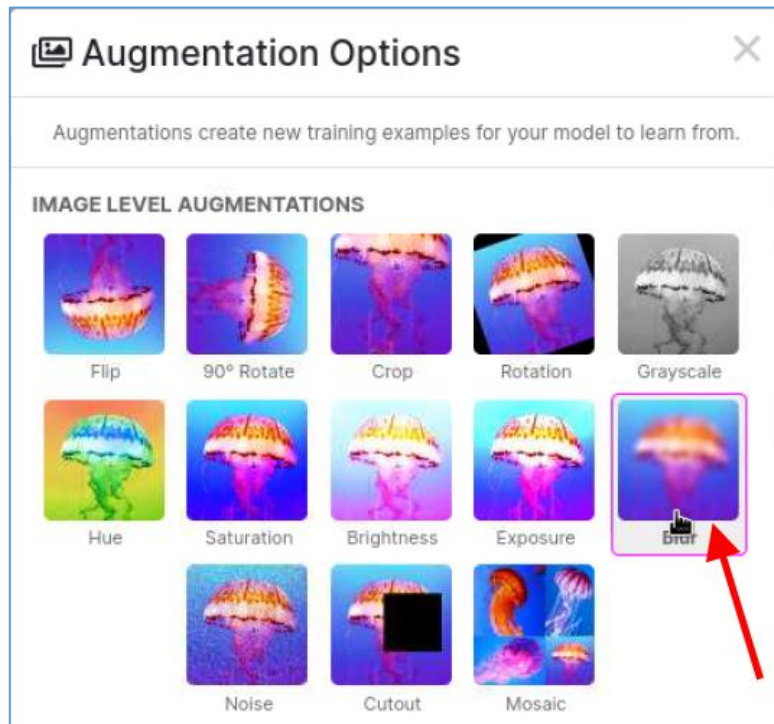
2.42 ปรับการเอียงรูปภาพในแนวตั้งและแนวนอน จากนั้นคลิกปุ่ม Apply



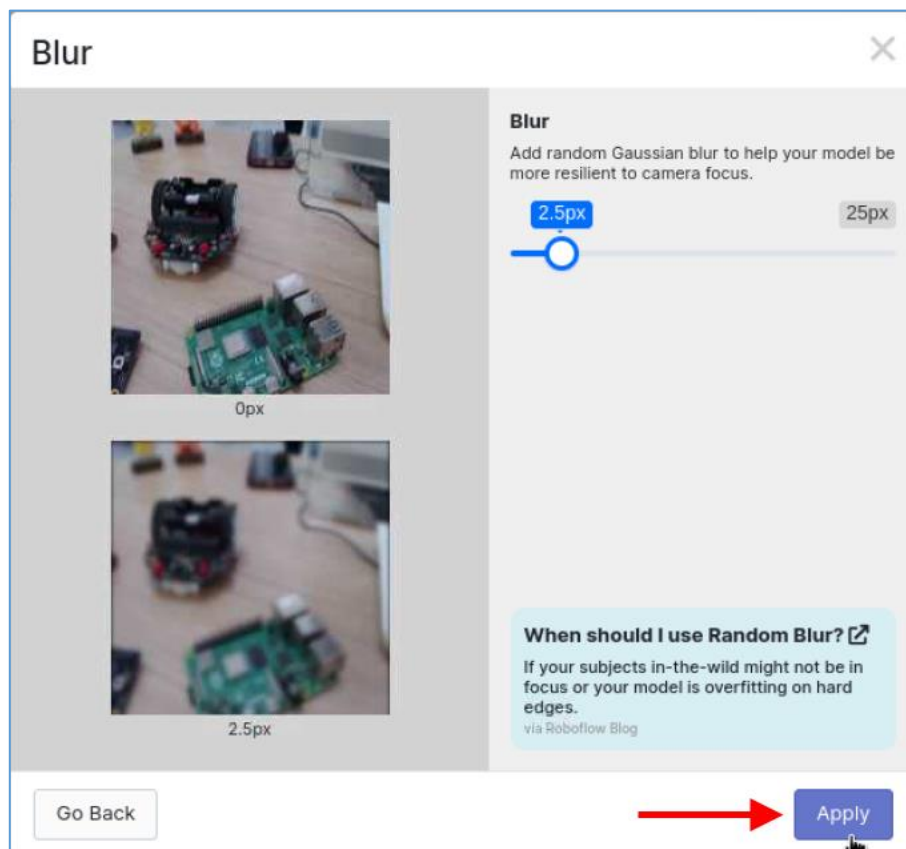
2.43 หากต้องการเพิ่ม Augmentation ให้คลิก Add Augmentation Step



2.44 ในที่นี้จะเลือกเป็นแบบ Blur



2.45 ปรับการเบลอของภาพ จากนั้นคลิกปุ่ม Apply





2.46 คลิกรูป Continue

4

### Augmentation

[? What can augmentation do?](#)

Create new training examples for your model to learn from by generating augmented versions of each image in your training set.

<b>Shear</b> ±15° Horizontal, ±15° Vertical	<a href="#">Edit</a>	×
<b>Blur</b> Up to 2.5px	<a href="#">Edit</a>	×

[+ Add Augmentation Step](#)

Continue

←

2.47 คลิกรูป Generate

✓

**Preprocessing**

**Auto-Orient:** Applied  
**Resize:** Stretch to 640×640

✓

**Augmentation**

**Shear:** ±15° Horizontal, ±15° Vertical  
**Blur:** Up to 2.5px

5

**Generate**

Review your selections and select a version size to create a moment-in-time snapshot of your dataset with the applied transformations.

Larger versions take longer to train but often result in better model performance. [See how this is calculated >>](#)

**Maximum Version Size**

60 images (3x) ▼

Generate

←

## 2.48 ส่งออก Dataset โดยคลิกปุ่ม Export Dataset

aset

v2 2023-09-18 10:57am  
Generated on Sep 18, 2023

**This version doesn't have a model.**

Train an optimized, state of the art model with Roboflow or upload a custom trained model to use features like Label Assist and Model Evaluation and deployment options like our auto-scaling API and edge device support.

Train with Roboflow Custom Train and Upload

Available Credits: 3

## 2.49 เลือกรูปแบบการส่งออกเหมือนครั้งแรก จากนั้นคลิกปุ่ม Continue

Export

Format

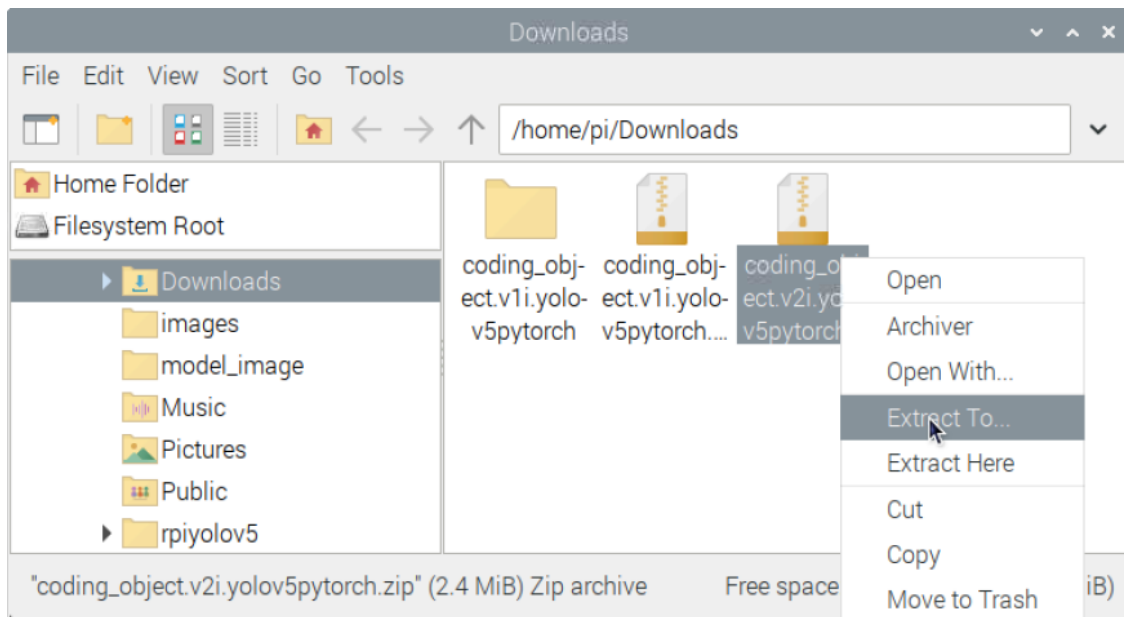
YOLO v5 PyTorch

TXT annotations and YAML config used with YOLOv5.

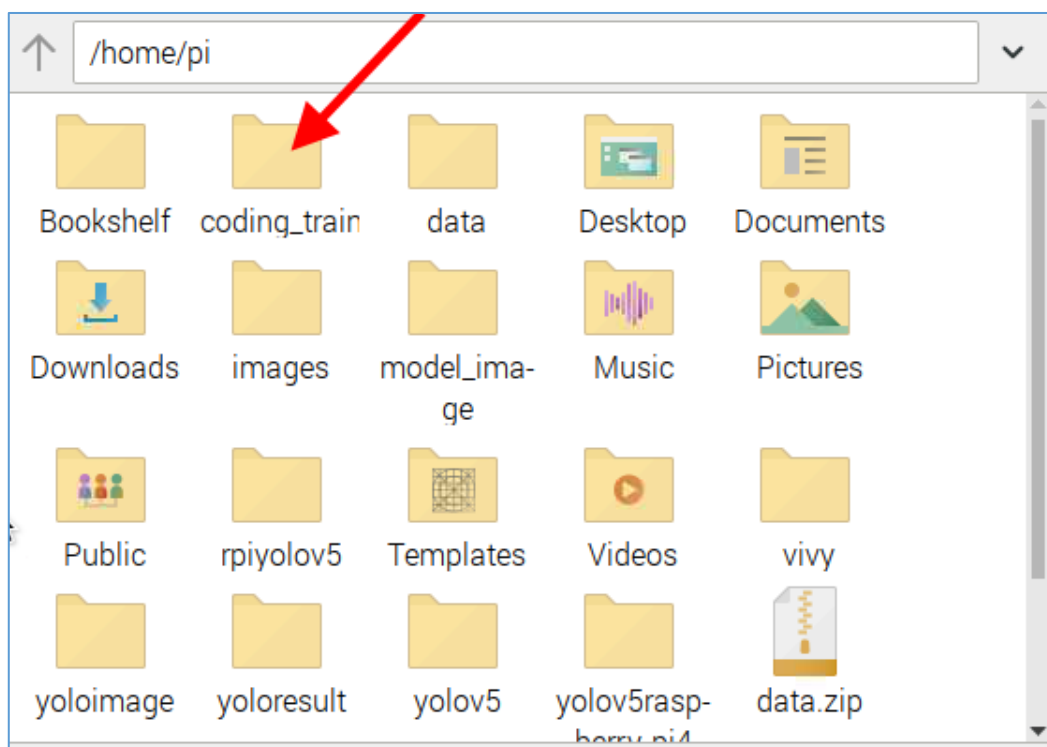
download zip to computer  show download code

Cancel Continue

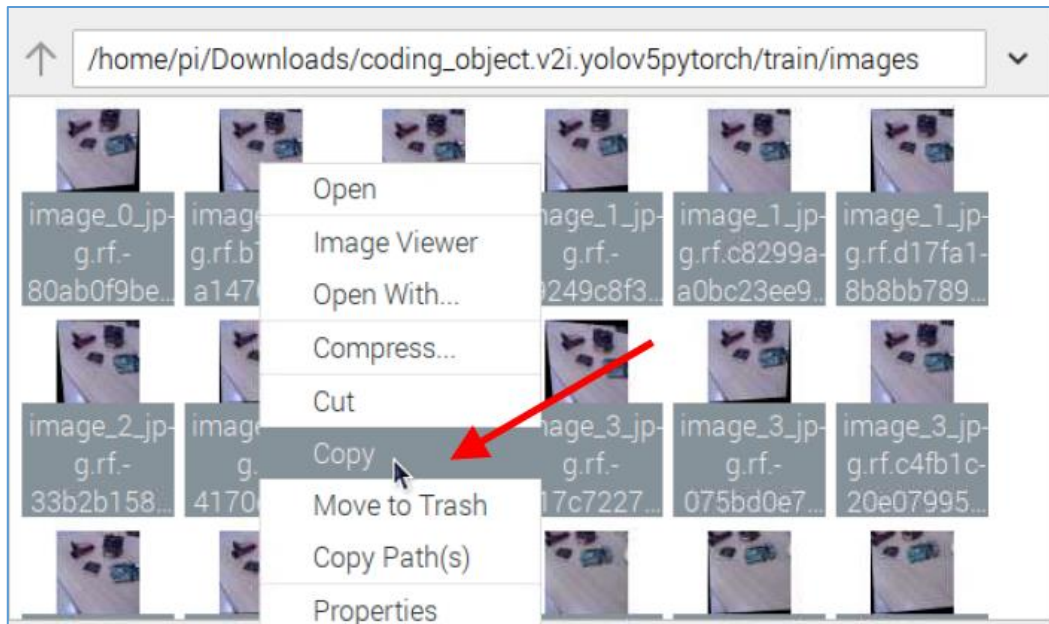
2.50 ไฟล์ที่สองจะถูกดาวน์โหลดมาไว้ในโฟลเดอร์ Downloads เช่นเดียวกับไฟล์แรก ให้ทำการแตกไฟล์เช่นเดียวกัน โดยคลิกขวาเลือก Extract To...



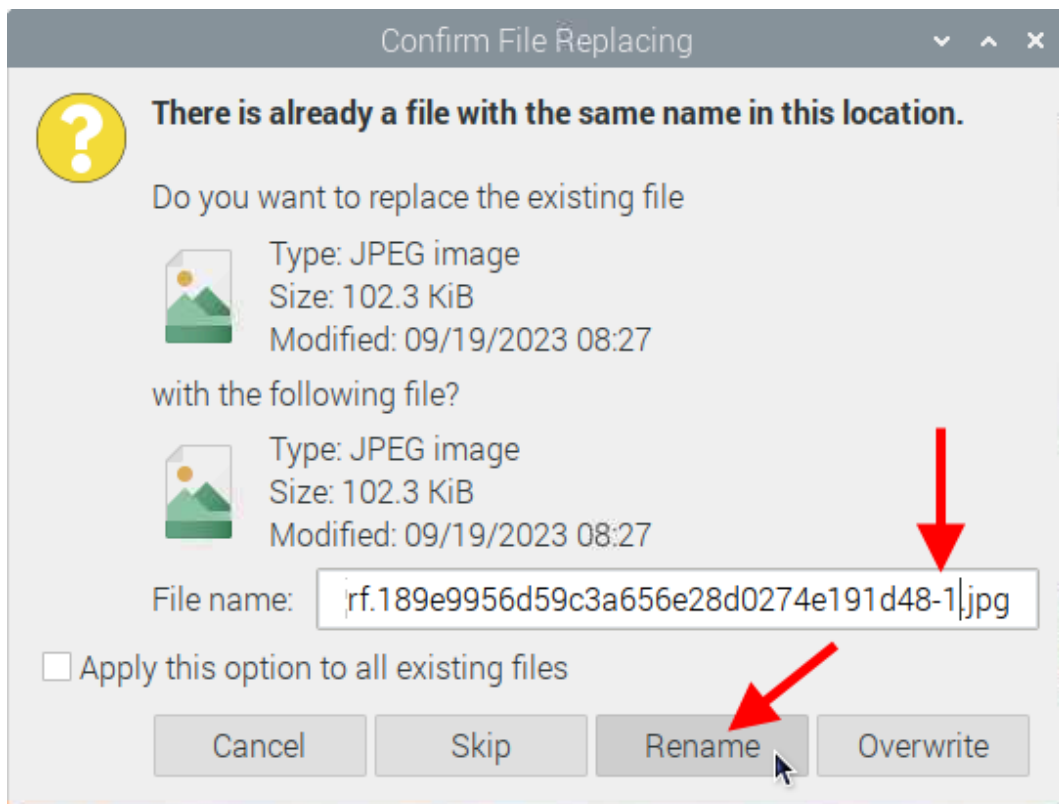
2.51 คัดลอกโฟลเดอร์ train ที่อยู่ใน /home/pi/Downloads/coding\_object.v1i.yolov5pytorch ไปวางที่ /home/pi จากนั้นเปลี่ยนชื่อโฟลเดอร์เป็น coding\_train



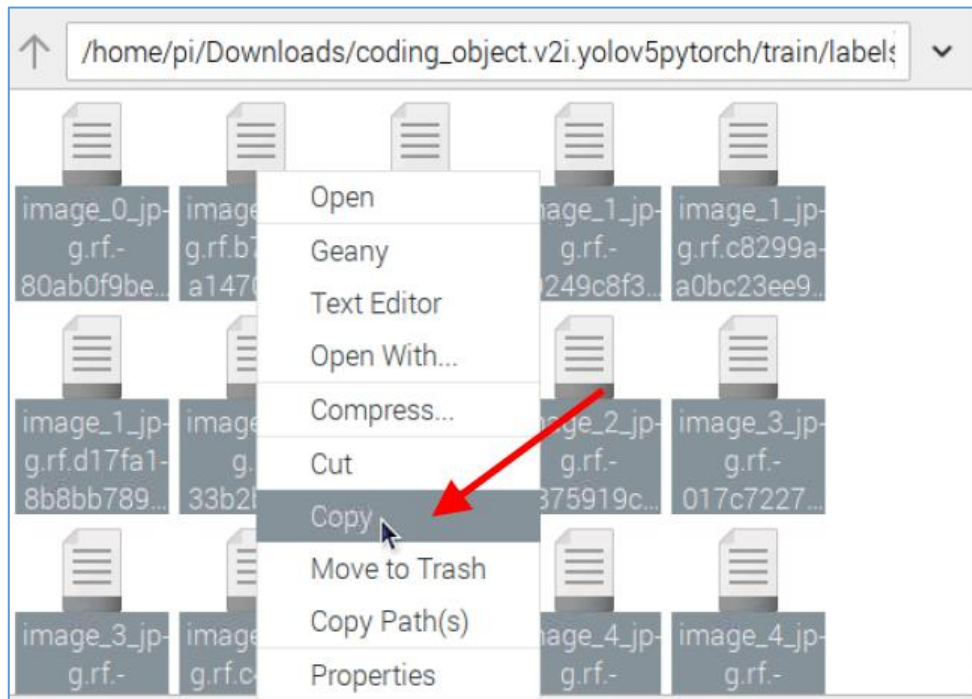
2.52 คัดลอกรูปภาพใน /home/pi/Downloads/coding\_object.v2i.yolov5pytorch/train/images



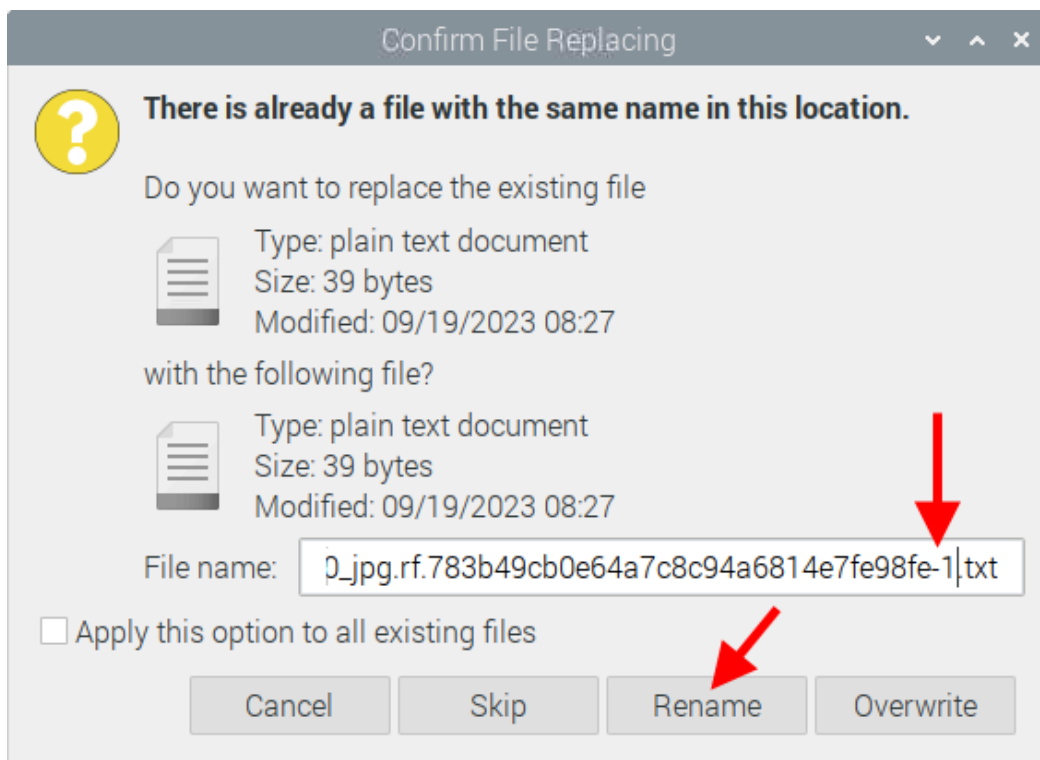
2.53 วางรูปภาพในโฟลเดอร์ `/home/pi/coding_train/images` หากมีชื่อไฟล์ซ้ำกับชื่อไฟล์เดิม ให้เปลี่ยนชื่อโดยใส่ “-1” ไว้ด้านหลังชื่อไฟล์ แล้วคลิก Rename



2.54 คัดลอกไฟล์เลเบล .txt ใน /home/pi/Downloads/coding\_object.v2i.yolov5pytorch/train/labels

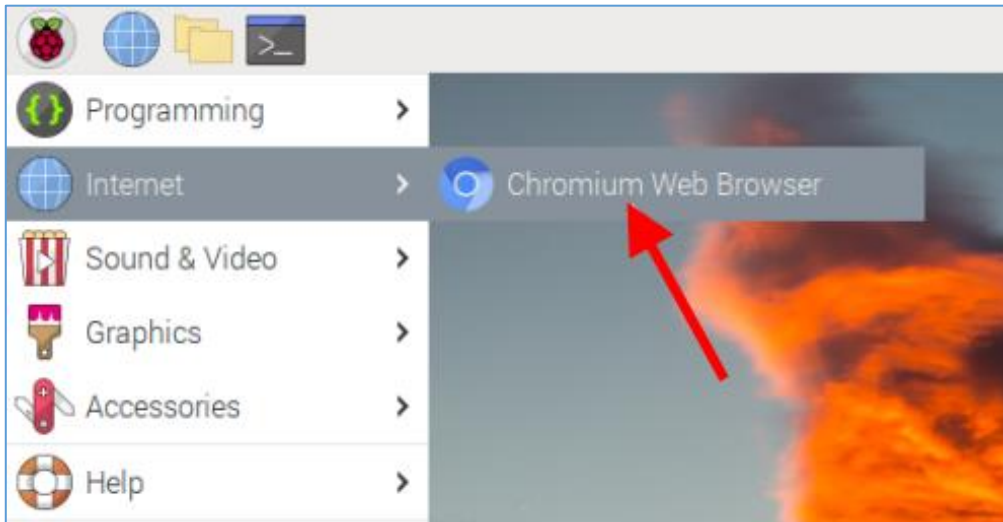


2.55 วางไฟล์ .txt ไว้ในโฟลเดอร์ /home/pi/coding\_train/labels หากมีชื่อไฟล์ซ้ำกับชื่อไฟล์เดิม ให้เปลี่ยนชื่อโดยใส่ "-1" ไว้ด้านหลังชื่อไฟล์ แล้วคลิก Rename

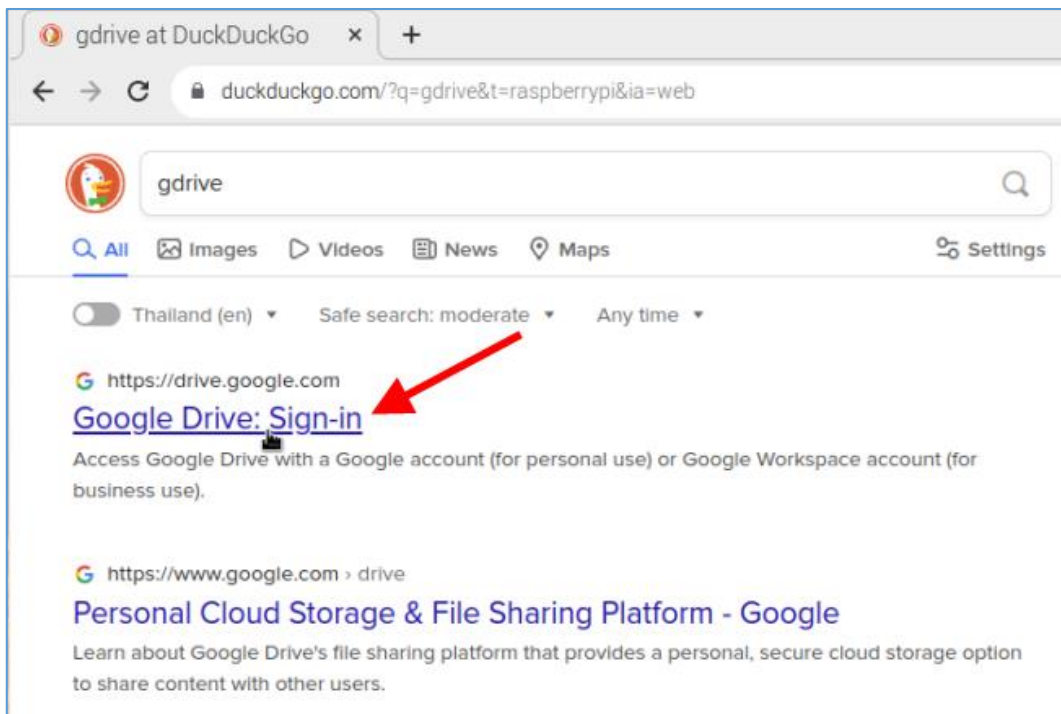


### ขั้นตอนที่ 3 อัปโหลดชุดข้อมูล (Dataset) ขึ้น Google Drive

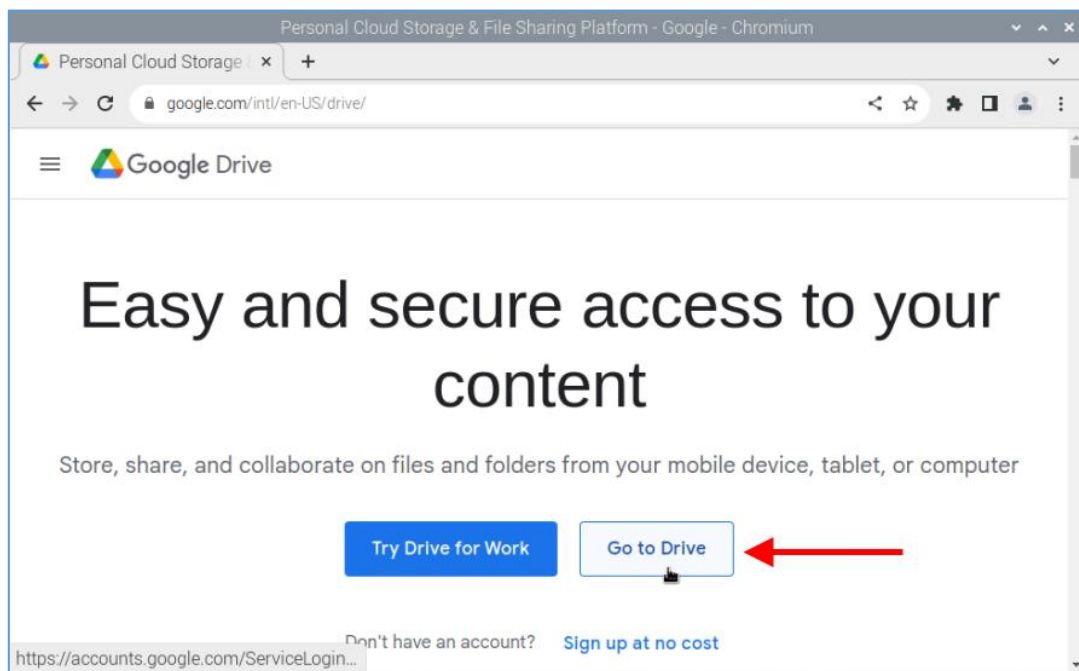
#### 3.1 เปิดโปรแกรม Chromium Web Browser



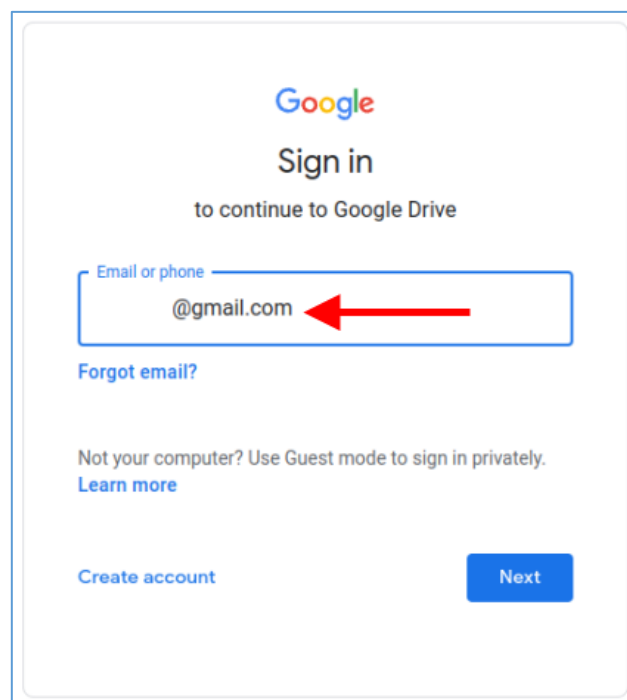
#### 3.2 พิมพ์คำค้นหาว่า “gdrive” จากนั้นคลิกเข้าไปยังเว็บไซต์ Google Drive



### 3.3 คลิกปุ่ม Go to Drive

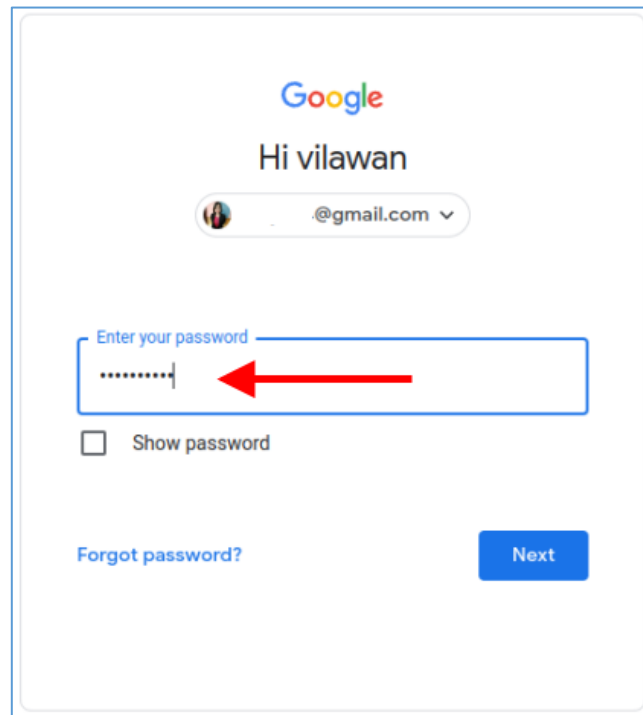


### 3.4 กรอกอีเมล จากนั้นคลิกปุ่ม Next






### 3.5 กรอกรหัสผ่าน จากนั้นคลิกปุ่ม Next



Google

Hi vilawan

 @gmail.com ▾

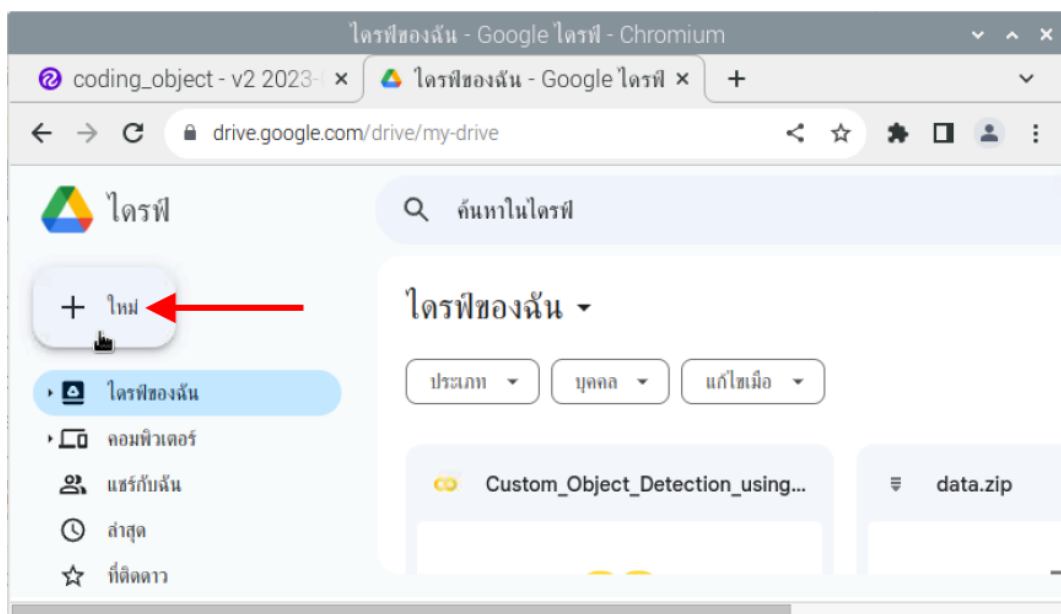
Enter your password

..... | ←

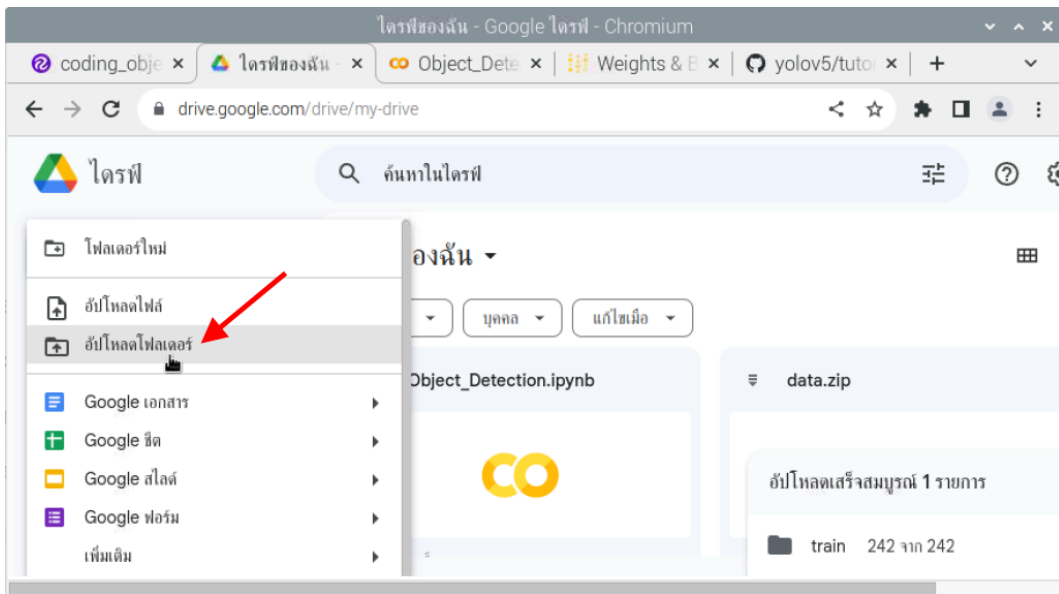
Show password

[Forgot password?](#) [Next](#)

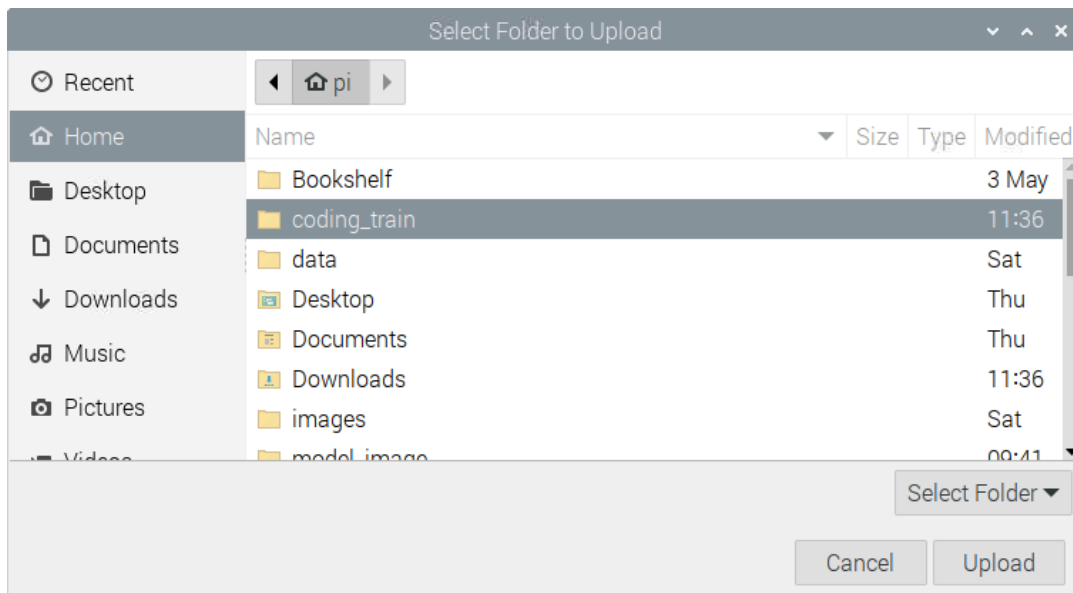
### 3.6 อัปโหลดไฟล์เดอร์ Dataset ขึ้น Google Drive โดยคลิกปุ่ม + ใหม่



### 3.7 คลิก อัปโหลดโฟลเดอร์

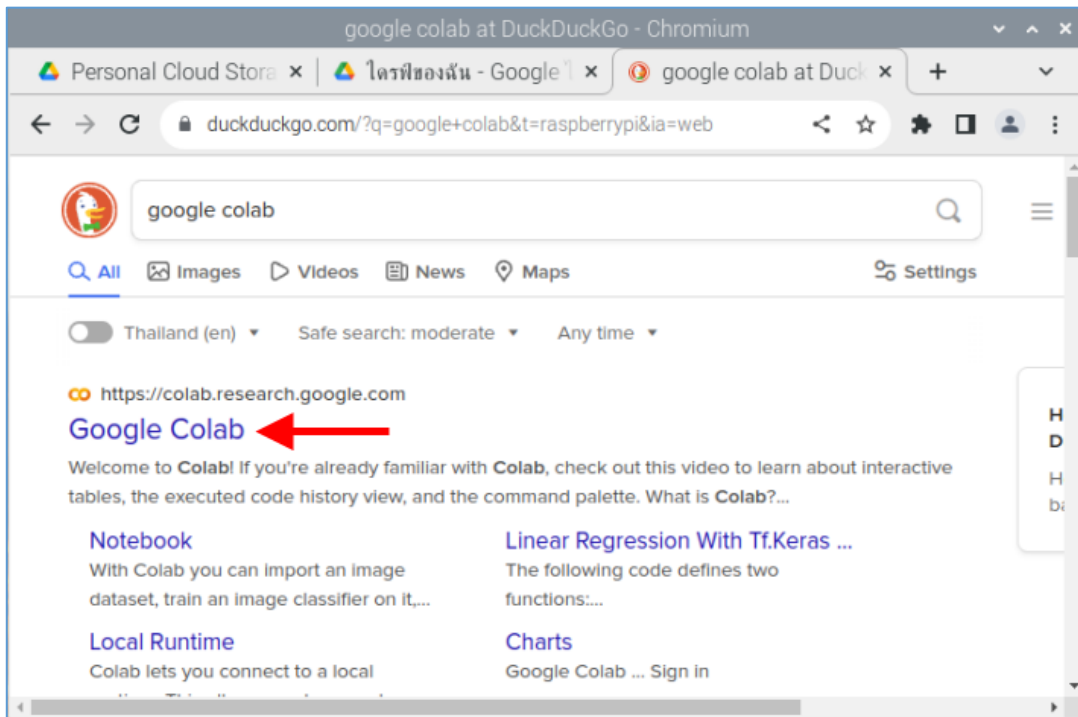


### 3.8 เลือกโฟลเดอร์ Dataset ในที่นี้คือโฟลเดอร์ชื่อ coding\_train จากนั้นคลิกปุ่ม Upload

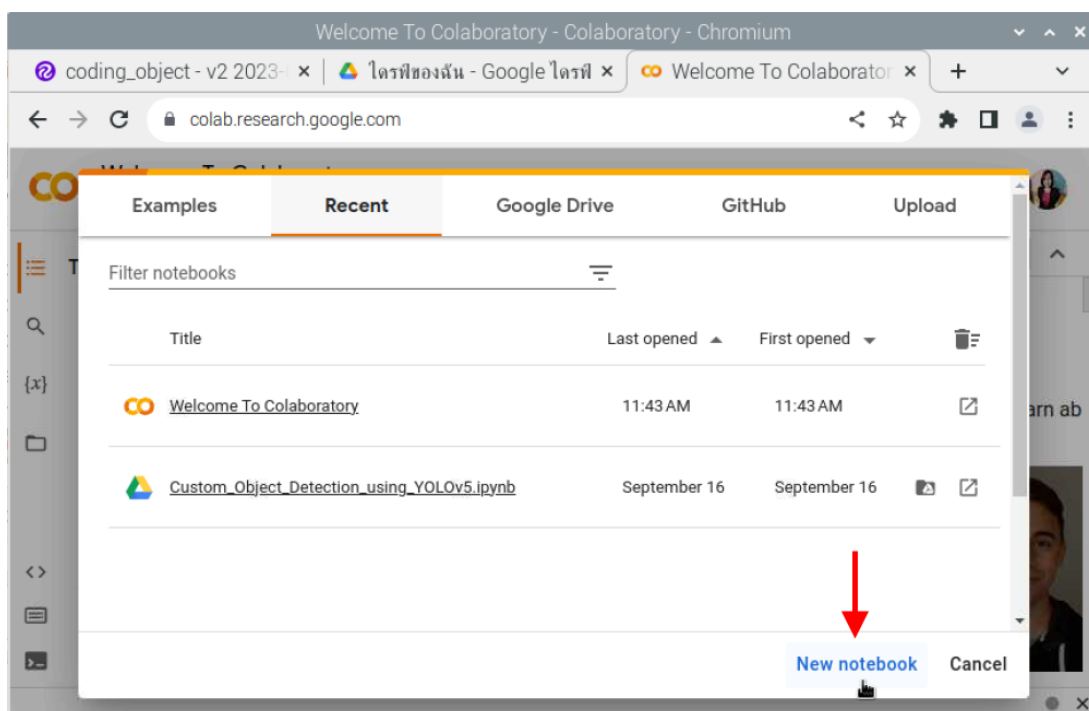


## ขั้นตอนที่ 4 ฝึกฝนโมเดล (Train) ด้วย Google Colab

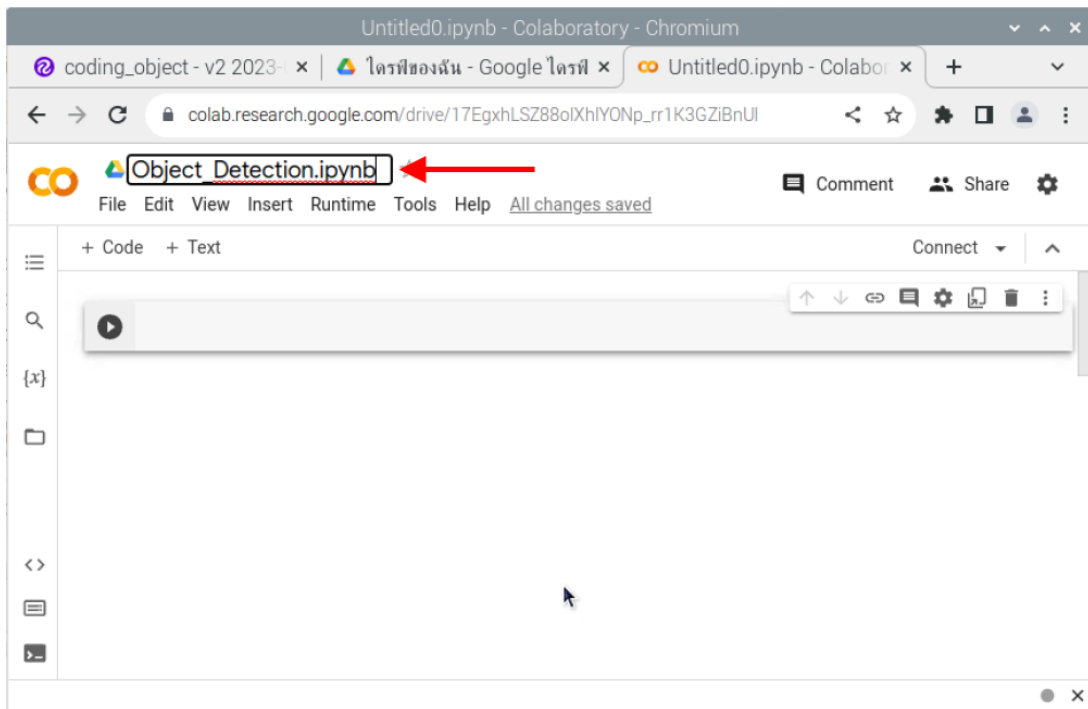
4.1 คลิกเปิดแท็บใหม่ และพิมพ์คำค้นหาว่า “google colab” จากนั้นคลิกเข้าเว็บไซต์ Google Colab



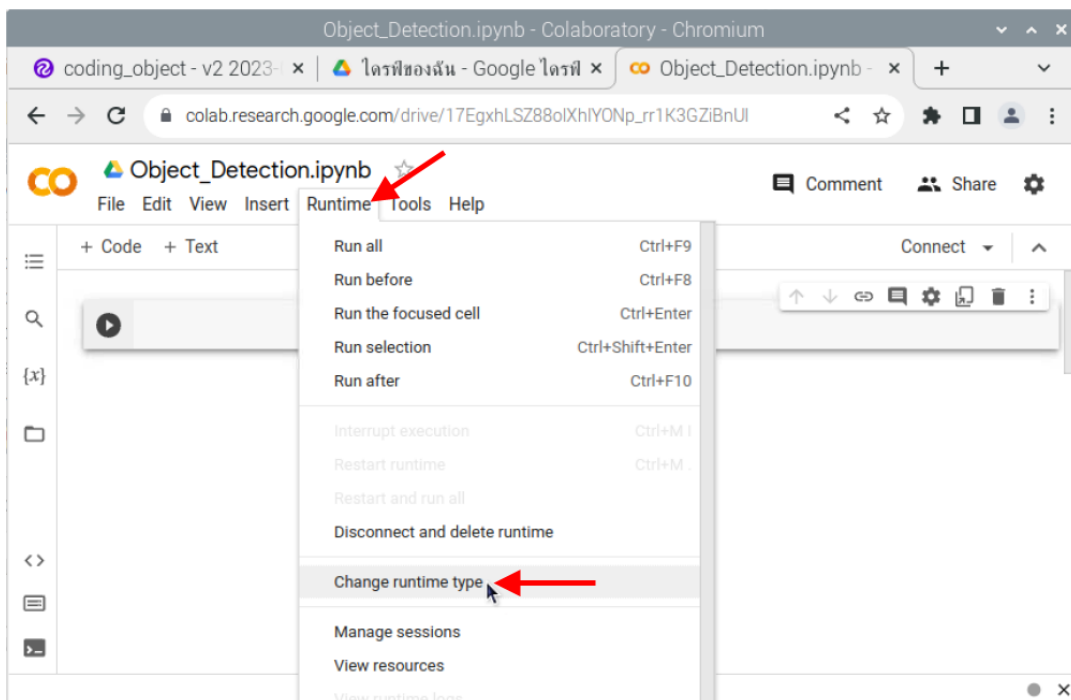
4.2 สร้างพื้นที่การทำงานใหม่ โดยคลิกปุ่ม New notebook หรือคลิกแท็บ File --> New notebook



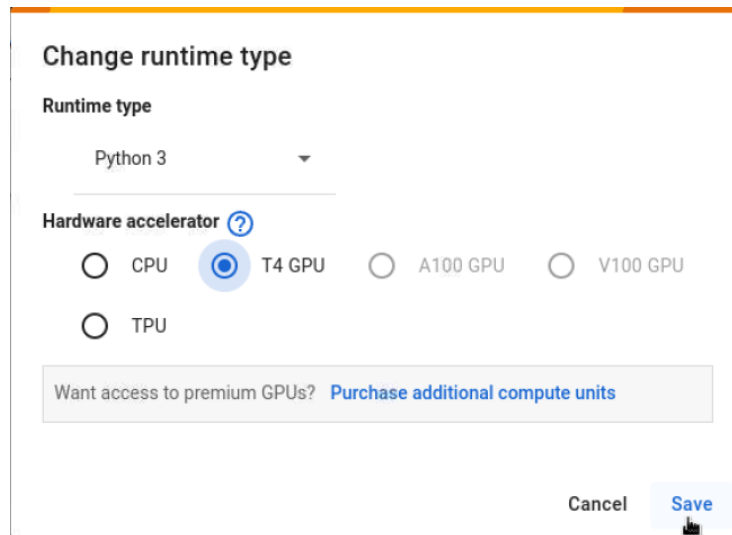
#### 4.3 ตั้งชื่อโมเดล



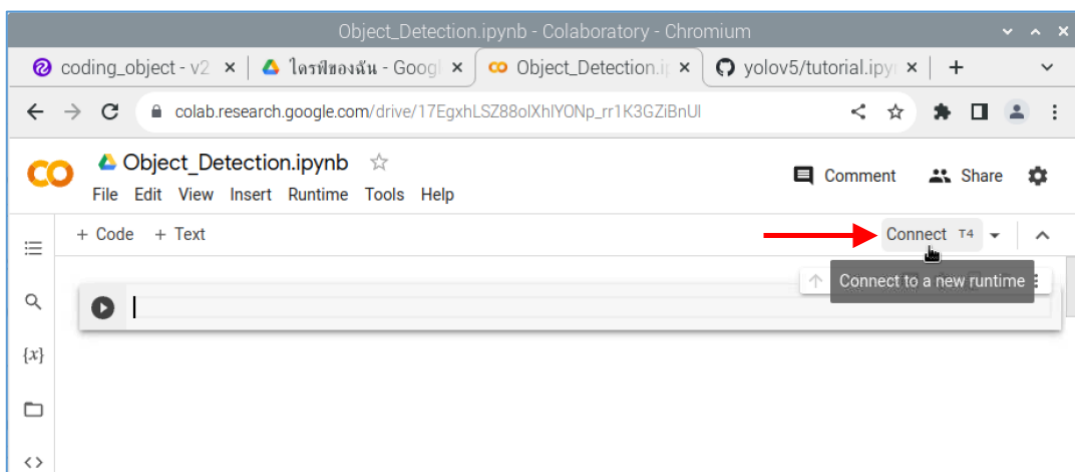
#### 4.4 เลือกรันใหม่ โดยคลิกที่ Runtime แล้วคลิกเลือก Change runtime type



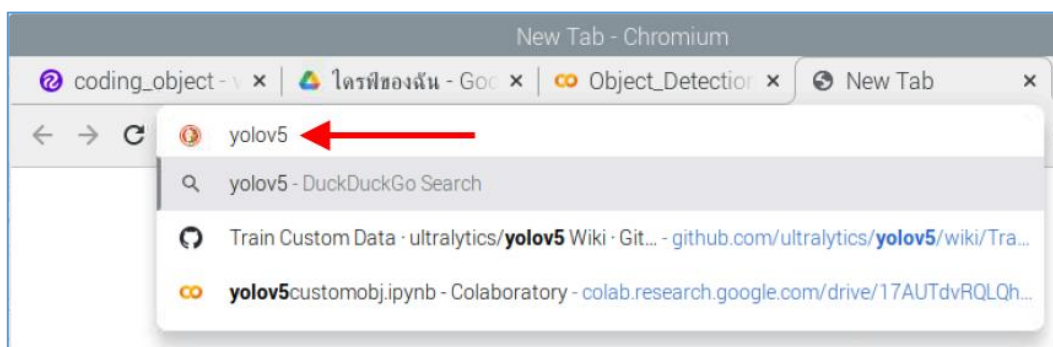
4.5 เลือกรันไทม์แบบ GPU เพื่อให้ประมวลผลเร็วขึ้น จากนั้นคลิกปุ่ม **Save**



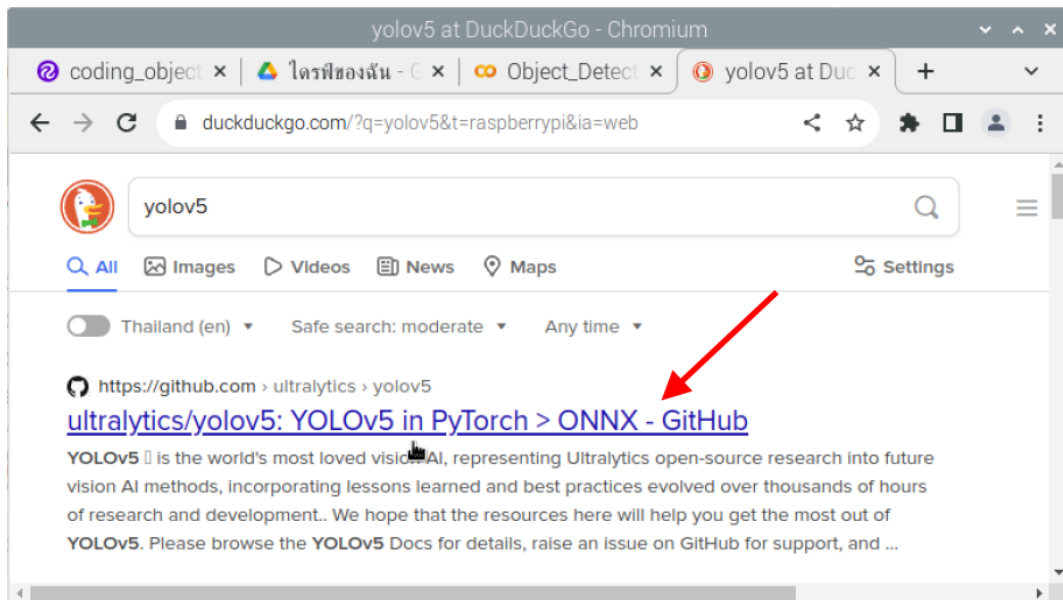
4.6 คลิกปุ่ม **Connect** เพื่อเชื่อมต่อกับบัญชี Gmail



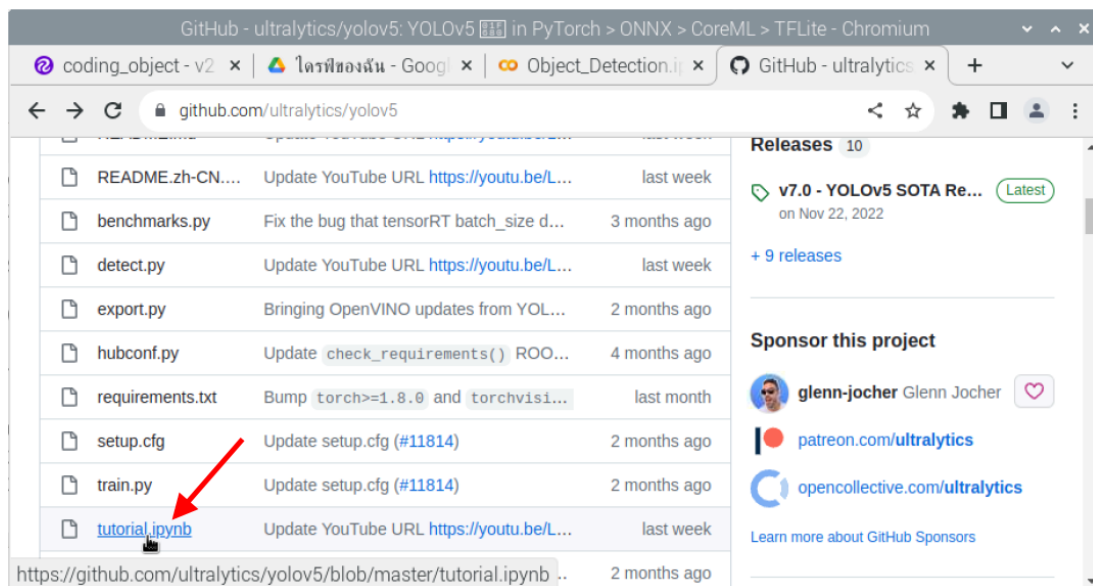
4.7 ค้นหาตัวอย่างโค้ดในการรันโปรแกรม โดยเปิดแท็บใหม่ และพิมพ์คำค้นหาว่า “yolov5”



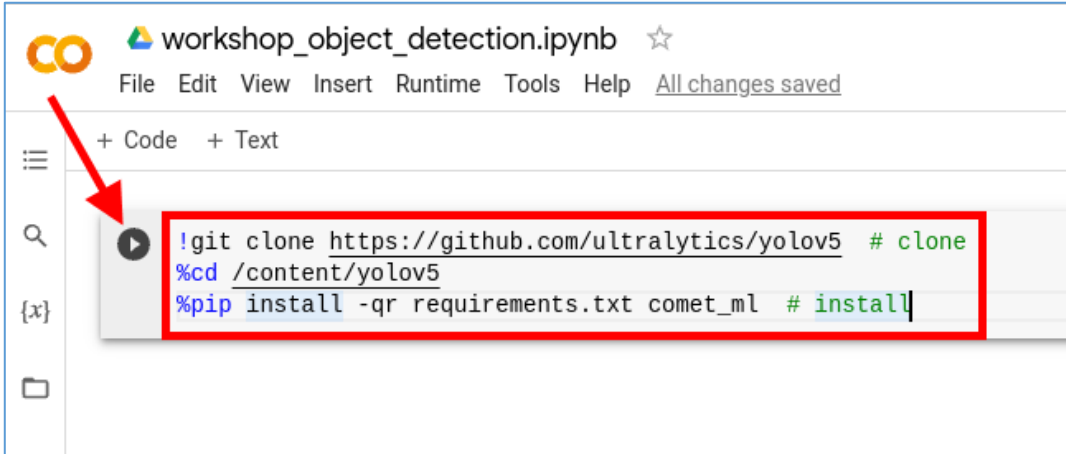
#### 4.8 คลิกเข้าเว็บไซต์ <https://github.com/ultralytics/yolov5>



#### 4.9 คลิกไฟล์ [tutorial.ipynb](#)

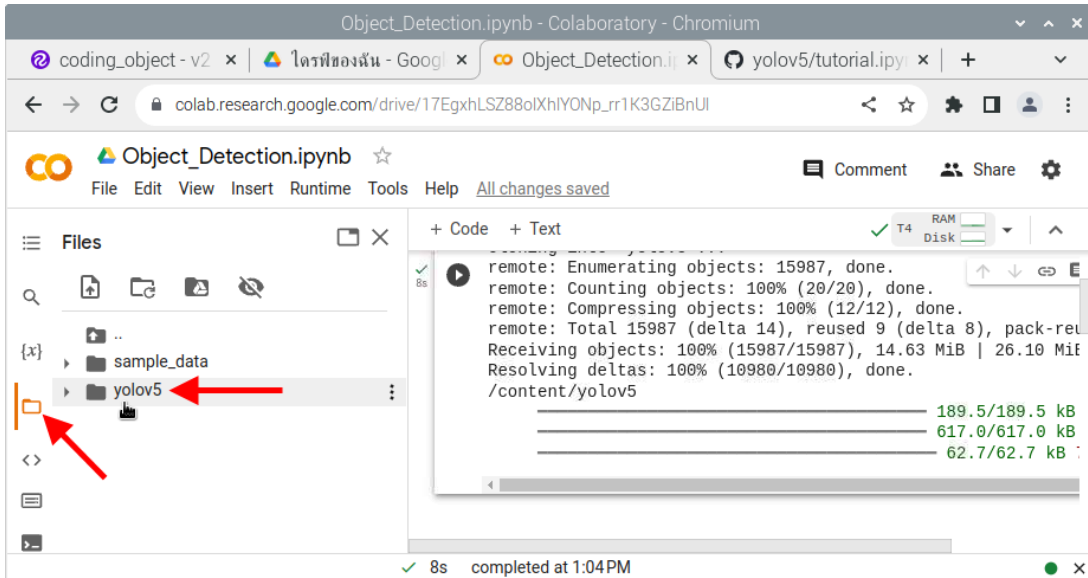


4.10 คัดลอกโค้ดคำสั่งเพื่อดาวน์โหลดไฟล์ Source Code ของ yolov5 และติดตั้งไลบรารีต่าง ๆ ที่จำเป็นต้องใช้ จากนั้นคลิกปุ่มรัน ดังรูป



```
!git clone https://github.com/ultralytics/yolov5 # clone
%cd /content/yolov5
%pip install -qr requirements.txt comet_ml # install
```

4.11 คลิกที่ Files (รูปโฟลเดอร์ด้านซ้ายมือ) จะปรากฏโฟลเดอร์ yolov5 ขึ้นมา

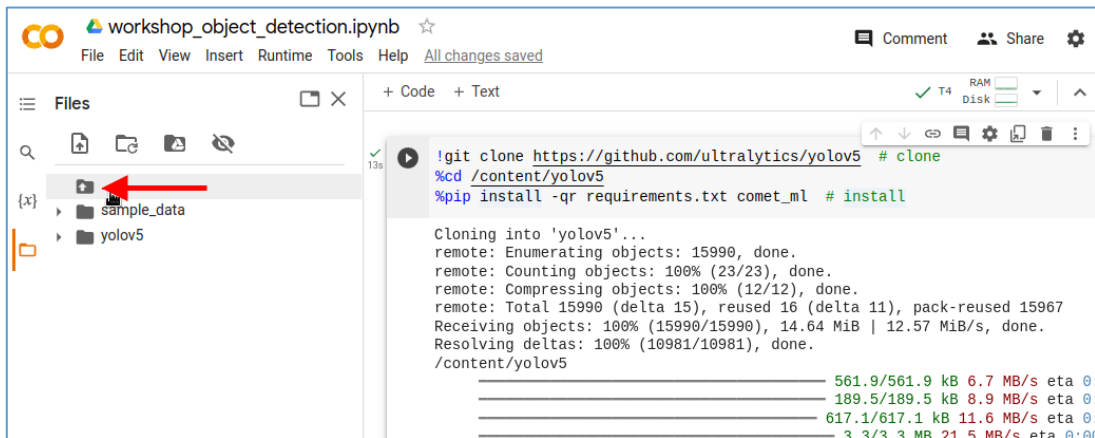


```
remote: Enumerating objects: 15987, done.
remote: Counting objects: 100% (20/20), done.
remote: Compressing objects: 100% (12/12), done.
remote: Total 15987 (delta 14), reused 9 (delta 8), pack-reu
Receiving objects: 100% (15987/15987), 14.63 MiB | 26.10 Mi
Resolving deltas: 100% (10980/10980), done.
/content/yolov5
_____ 189.5/189.5 kB
_____ 617.0/617.0 kB
_____ 62.7/62.7 kB
```

8s completed at 1:04 PM



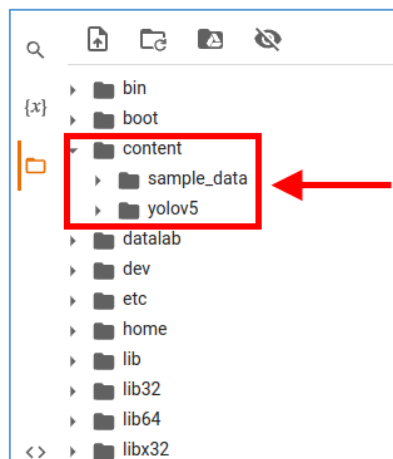
#### 4.12 คลิก จะเห็นว่าโฟลเดอร์ yolov5 อยู่ในโฟลเดอร์ content



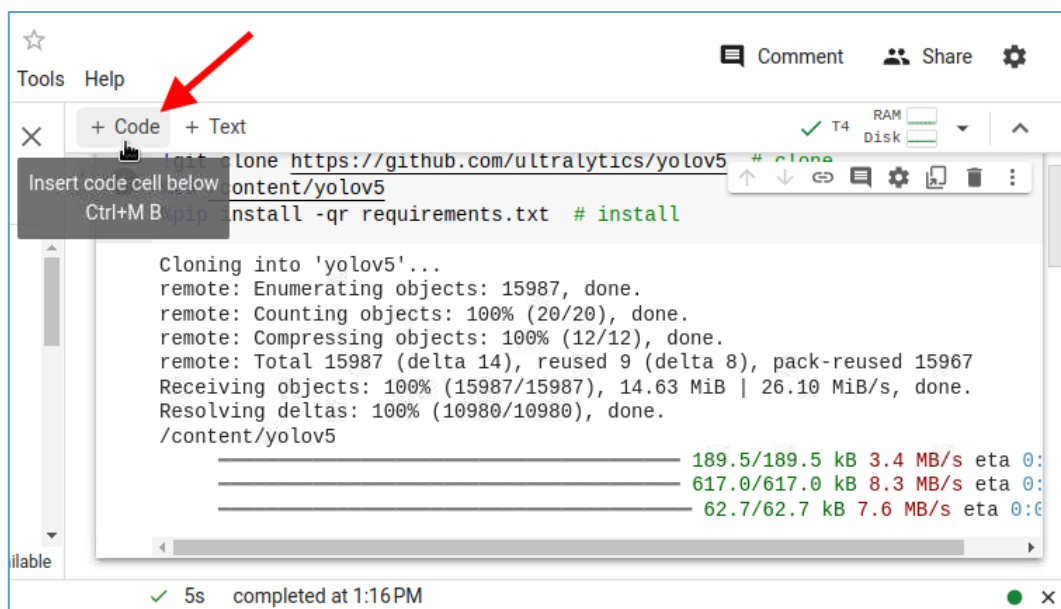
The screenshot shows the JupyterLab interface for a notebook named 'workshop\_object\_detection.ipynb'. On the left, the 'Files' panel shows a directory structure with 'sample\_data' and 'yolov5' folders. A red arrow points to the 'yolov5' folder. The main terminal area shows the execution of the following commands:

```
!git clone https://github.com/ultralytics/yolov5 # clone
%cd /content/yolov5
%pip install -qr requirements.txt comet_ml # install
```

The terminal output shows the cloning process into 'yolov5' and the installation of dependencies. A progress bar at the bottom indicates the download status of various files.



#### 4.13 ทดสอบว่าการติดตั้ง PyTorch สมบูรณ์หรือไม่ โดยคลิก + Code เพื่อเพิ่มโค้ดบรรทัดใหม่



The screenshot shows the JupyterLab interface with the '+ Code' button highlighted by a red arrow. A tooltip above the button reads 'Insert code cell below Ctrl+M B'. The terminal area shows the execution of the same commands as in the previous screenshot:

```
!git clone https://github.com/ultralytics/yolov5 # clone
%cd /content/yolov5
%pip install -qr requirements.txt # install
```

The terminal output shows the cloning process into 'yolov5' and the installation of dependencies. A progress bar at the bottom indicates the download status of various files. The terminal output is partially obscured by the tooltip.

#### 4.14 คัดลอกคำสั่งจากไฟล์ `tutorial.ipnb` มาวาง จากนั้นคลิกปุ่มรัน ดังรูป

```

+ Code + Text
Resolving deltas: 100% (10980/10980), done.
[1] /content/yolov5
189.5/189.5 kB 3.4 MB/s eta 0:
617.0/617.0 kB 8.3 MB/s eta 0:
62.7/62.7 kB 7.6 MB/s eta 0:0

import torch
import utils
display = utils.notebook_init() # checks

5s completed at 1:16 PM

```

#### 4.15 จะได้ผลลัพธ์ดังรูป แสดงว่าติดตั้ง PyTorch ผ่าน

```

+ Code + Text
Resolving deltas: 100% (10980/10980), done.
[1] /content/yolov5
189.5/189.5 kB 3.4 MB/s eta 0:
617.0/617.0 kB 8.3 MB/s eta 0:
62.7/62.7 kB 7.6 MB/s eta 0:0

import torch
import utils
display = utils.notebook_init() # checks

YOLOv5 v7.0-218-g9e97ac3 Python-3.10.12 torch-2.0.1+cu118 CUDA:0 (Tesla T
Setup complete (2 CPUs, 12.7 GB RAM, 26.3/78.2 GB disk)

0s completed at 1:17 PM

```

4.16 ทดสอบการใช้ YOLOv5 ในการตรวจจับวัตถุของรูปภาพตัวอย่างที่มาพร้อมกับ yolov5 โดยรันไฟล์ detect.py และแสดงผลพร้อมออกมาด้านล่าง โดยคลิก + Code แล้วใส่คำสั่งดังรูป (คัดลอกโค้ดได้จาก tutorial.ipynb)

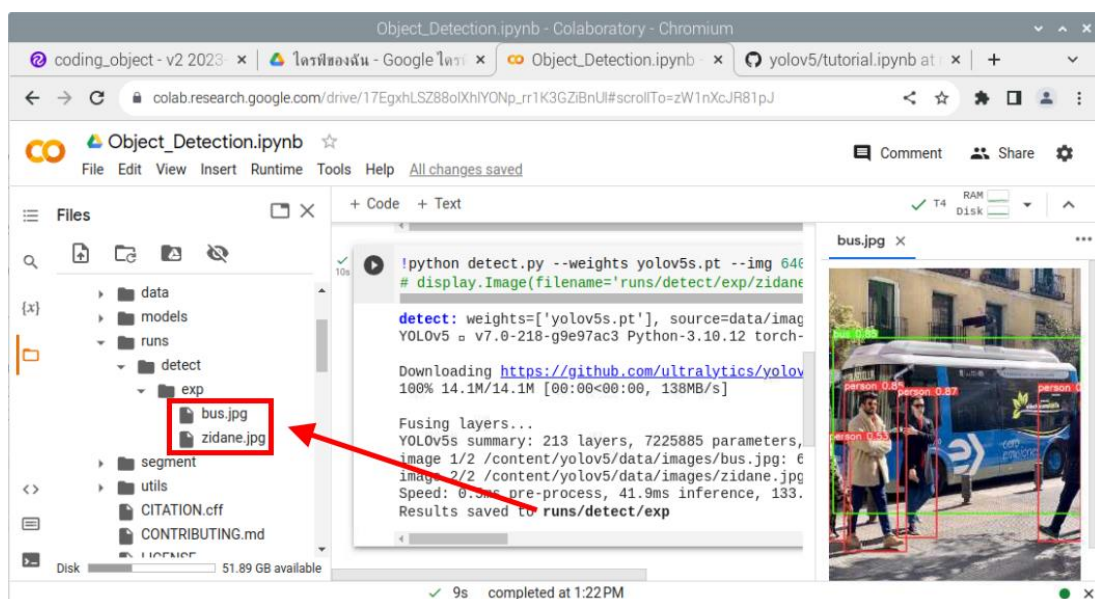
```
!python detect.py --weights yolov5s.pt --img 640 --conf 0.25 --source data/images
display.Image(filename='runs/detect/exp/zidane.jpg', width=600)
```

```
detect: weights=['yolov5s.pt'], source=data/images, data=data/coco128.yaml, imgsz=[640,
YOLOv5 v7.0-223-g005dcfa Python-3.10.12 torch-2.0.1+cu118 CUDA:0 (Tesla T4, 15102MiB)

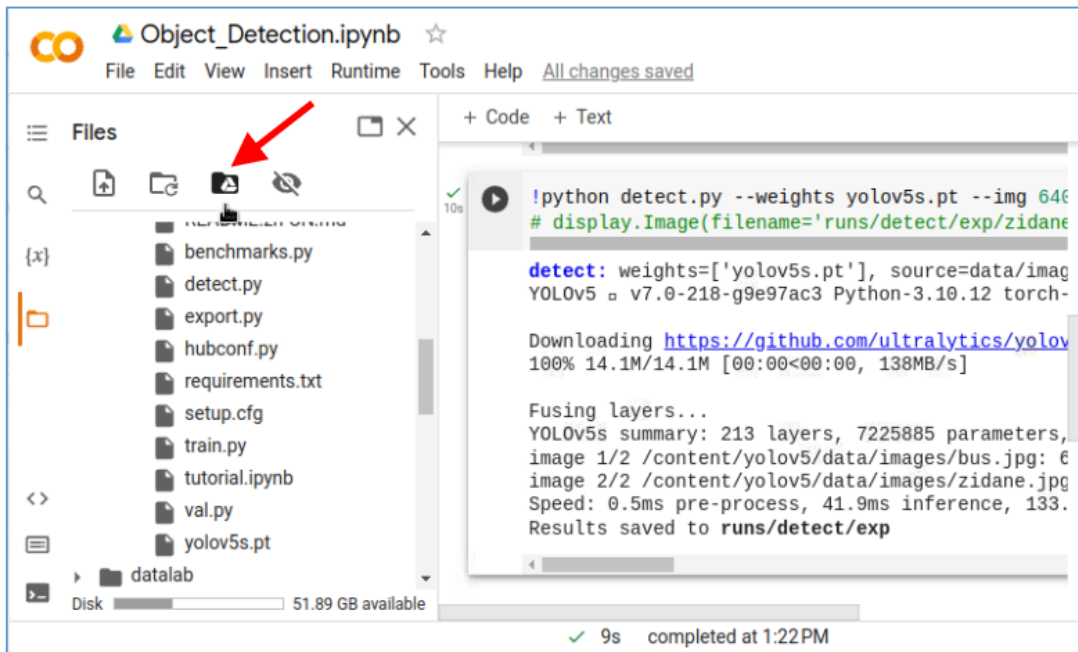
Downloading https://github.com/ultralytics/yolov5/releases/download/v7.0/yolov5s.pt to
100% 14.1M/14.1M [00:00<00:00, 75.1MB/s]

Fusing layers...
YOLOv5s summary: 213 layers, 7225885 parameters, 0 gradients
image 1/2 /content/yolov5/data/images/bus.jpg: 640x480 4 persons, 1 bus, 45.2ms
image 2/2 /content/yolov5/data/images/zidane.jpg: 384x640 2 persons, 2 ties, 44.5ms
Speed: 0.5ms pre-process, 44.8ms inference, 132.9ms NMS per image at shape (1, 3, 640,
Results saved to runs/detect/exp
```

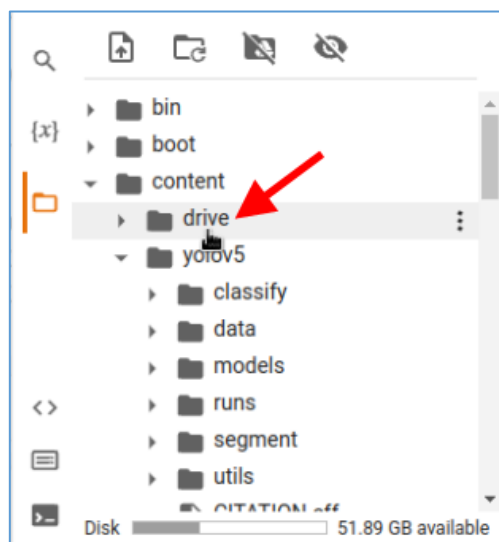
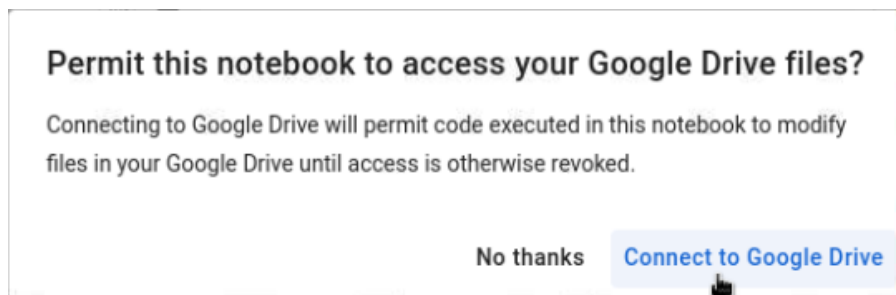
4.17 ผลลัพธ์จะถูกบันทึกไว้ในโฟลเดอร์ runs/detect/exp หากมีการรันไฟล์ detect อีกครั้ง จะได้โฟลเดอร์ exp1, exp2 ไปเรื่อย ๆ



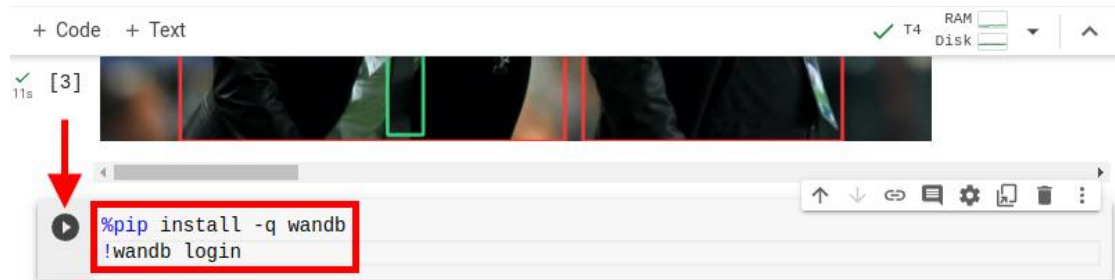
4.18 เชื่อมต่อกับ Google Drive เพื่อเรียกใช้งานรูปภาพ Dataset โดยคลิกที่ Mount Google Drive



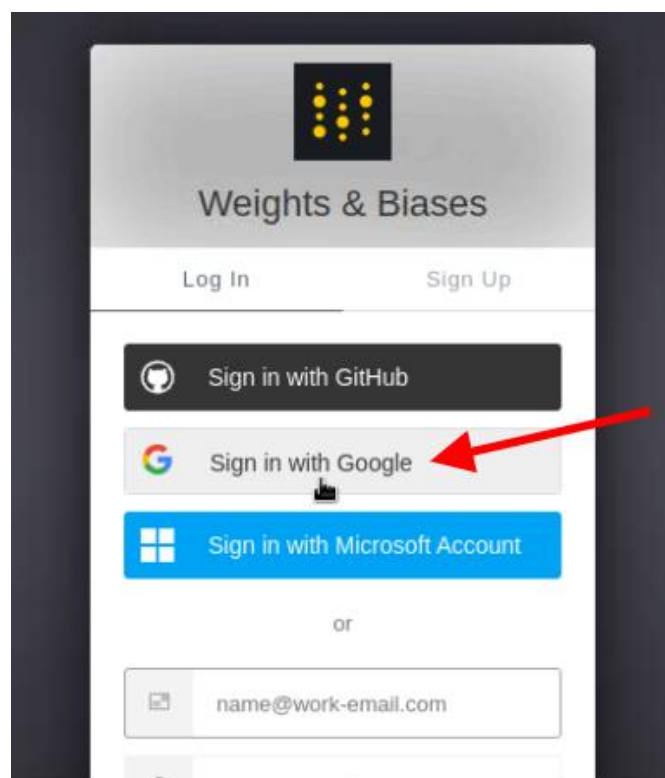
4.19 คลิก Connect to Google Drive จะปรากฏโฟลเดอร์ drive ขึ้นมาในโฟลเดอร์ Content



4.20 ติดตั้ง wandb (Weight & Biases) สำหรับใช้เป็น Dashboard ในการติดตามผลการสร้างแบบจำลอง โดยคลิก +Code แล้วพิมพ์คำสั่ง `%pip install -q wandb` และ `!wandb login` จากนั้นคลิกปุ่มรัน จะปรากฏลิงก์ขึ้นมาให้คลิกที่ <https://wandb.ai/authorize>




4.21 คลิกล็อกอินด้วย Gmail โดยคลิกปุ่ม [Sign in with Google](#)




## 4.22 เลือกบัญชีที่จะล็อกอิน

ลงชื่อเข้าใช้ด้วย Google

เลือกบัญชี  
เพื่อไปยัง [auth0.com](https://auth0.com)

 vilawan sornsangsen  
vivy284@gmail.com

 ใช้บัญชีอื่น

Google จะแชร์ชื่อ ที่อยู่อีเมล การตั้งค่าภาษา และรูปโปรไฟล์ของคุณ  
กับ auth0.com เพื่อดำเนินการต่อ

4.23 กรอกชื่อ / บริษัท จากนั้นคลิก [Sign Up](#)

## Create your account

Full name

Company or Institution

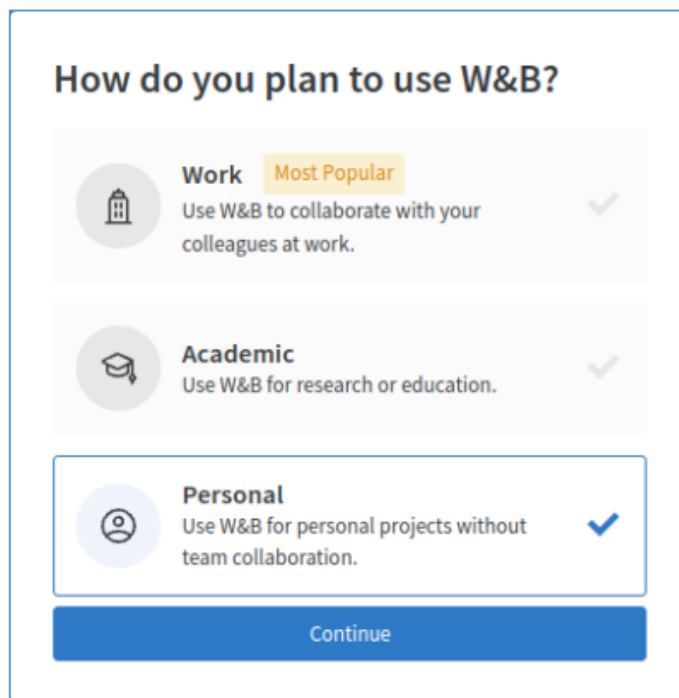
Username

Choose your username carefully! It cannot be changed after signup.

[Sign Up](#)

By clicking "Sign Up," you agree to W&B's terms & conditions.

4.24 เลือกแผนการใช้งาน ให้คลิกเป็น Personal จากนั้นคลิกปุ่ม Continue

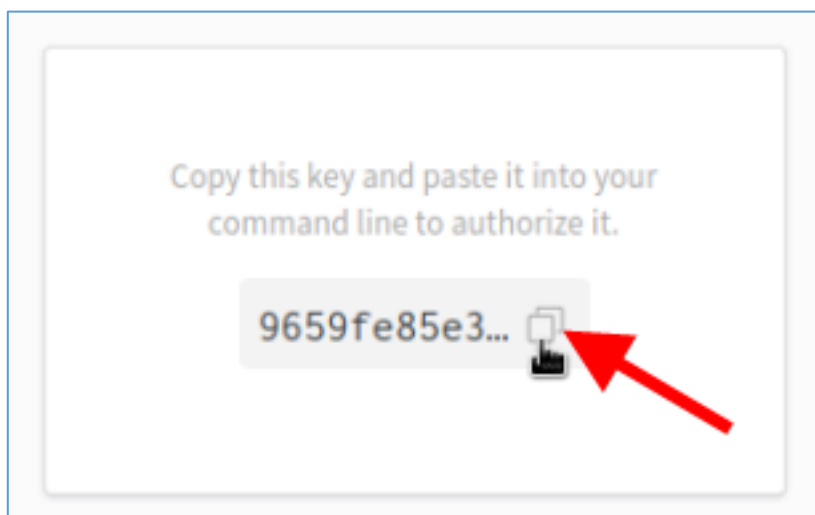


The screenshot shows a selection screen titled "How do you plan to use W&B?". It features three options, each with an icon, a title, a description, and a checkmark:

- Work** (Most Popular): Use W&B to collaborate with your colleagues at work.
- Academic**: Use W&B for research or education.
- Personal**: Use W&B for personal projects without team collaboration. This option is selected, indicated by a blue checkmark and a blue border around the card.

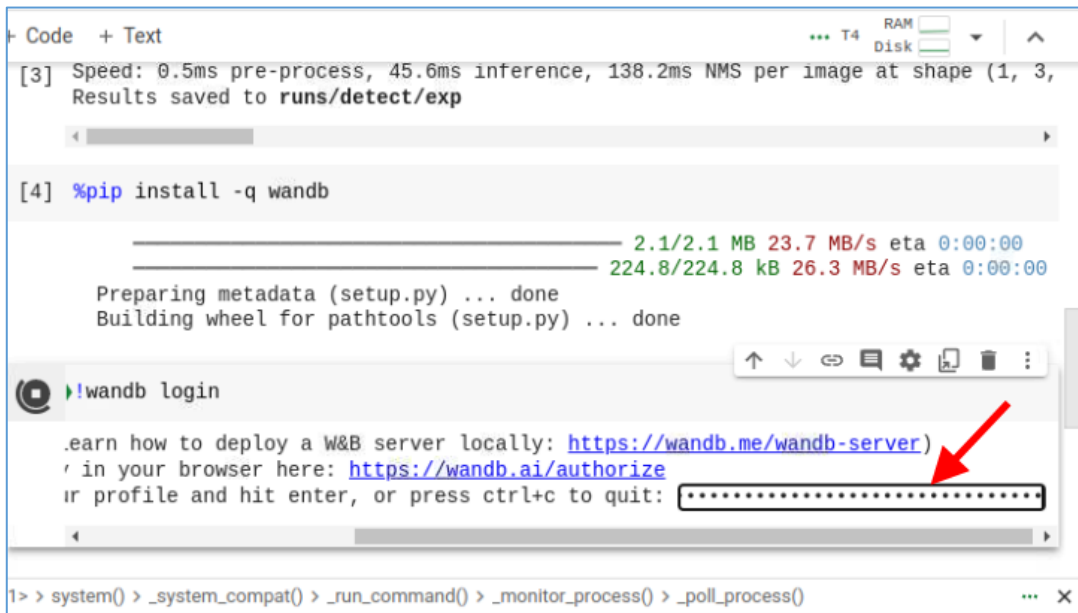
At the bottom of the screen is a blue button labeled "Continue".

4.25 คลิกคัดลอกรหัส Authorize ซึ่งแต่ละคนจะได้รหัสไม่เหมือนกัน





#### 4.26 นำรหัส Authorize มาวาง ดังรูป จากนั้น Enter



```

Code + Text
[3] Speed: 0.5ms pre-process, 45.6ms inference, 138.2ms NMS per image at shape (1, 3,
Results saved to runs/detect/exp

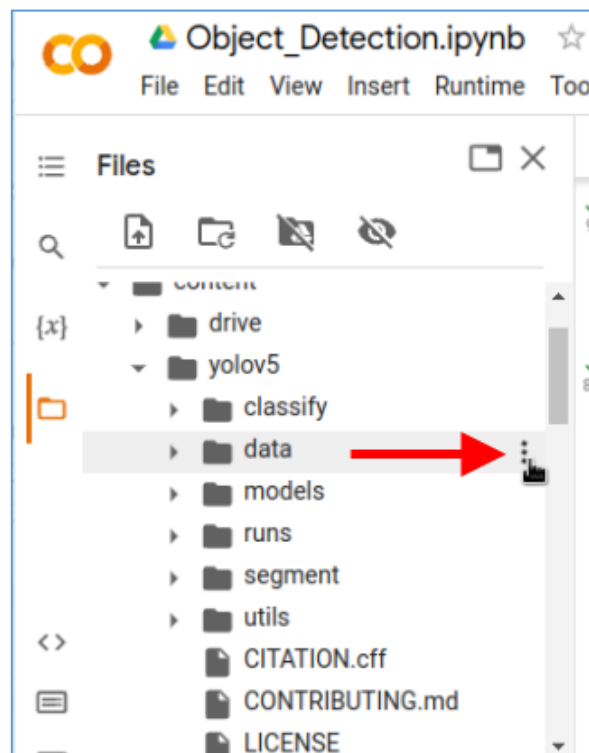
[4] %pip install -q wandb

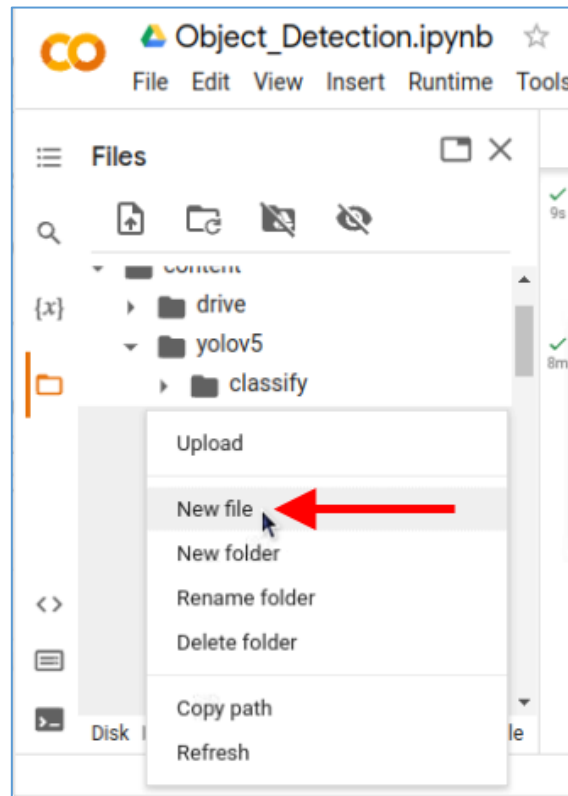
2.1/2.1 MB 23.7 MB/s eta 0:00:00
224.8/224.8 kB 26.3 MB/s eta 0:00:00
Preparing metadata (setup.py) ... done
Building wheel for pathtools (setup.py) ... done

!wandb login
Learn how to deploy a W&B server locally: https://wandb.me/wandb-server
or in your browser here: https://wandb.ai/authorize
or profile and hit enter, or press ctrl+c to quit:

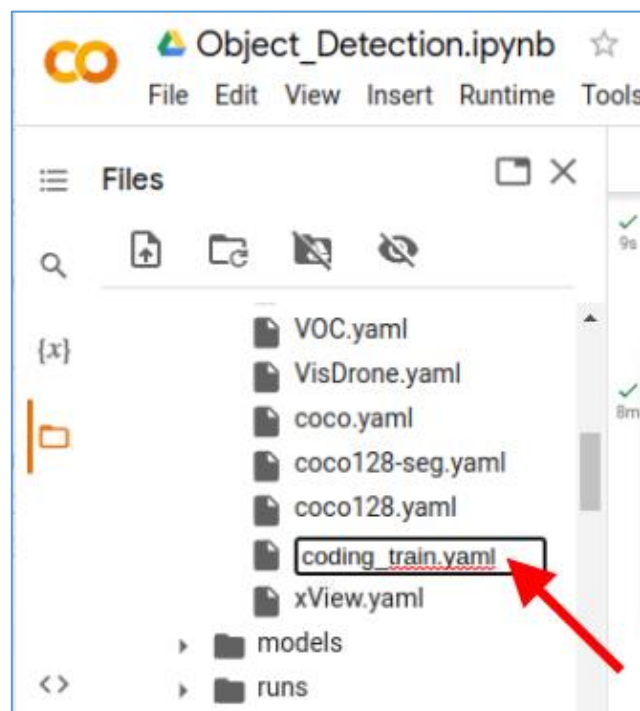
```

#### 4.27 สร้างไฟล์ใหม่ในโฟลเดอร์ data โดยคลิกที่จุด 3 จุดด้านหลังโฟลเดอร์ data

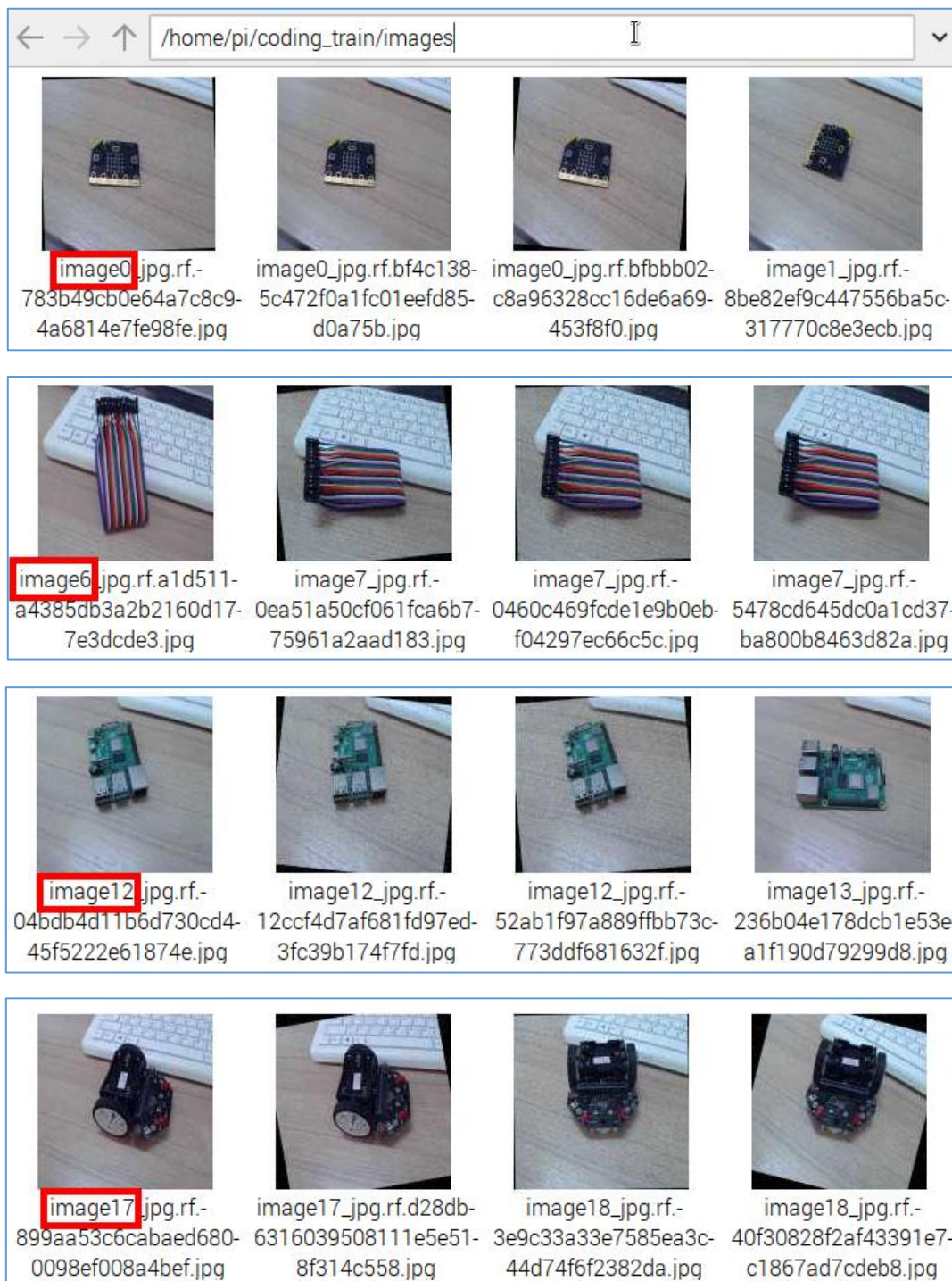


4.28 คลิกเลือก New file

## 4.29 ตั้งชื่อไฟล์เป็น coding\_train.yaml จากนั้นดับเบิ้ลคลิกที่ไฟล์เพื่อเขียนโค้ดคำสั่ง



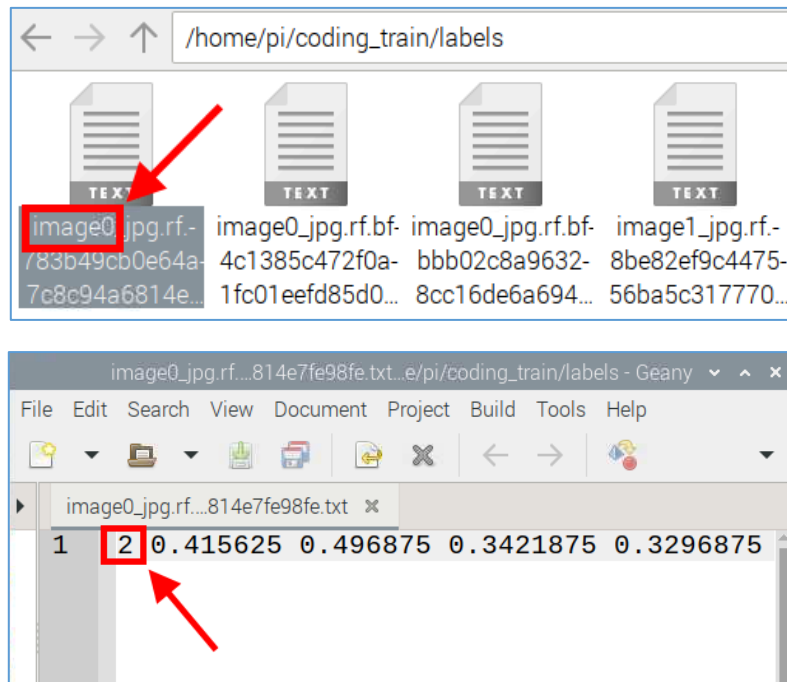
4.30 ตรวจสอบ id ของวัตถุทุกชิ้นที่จะทำการตรวจจับ โดยเปิดโฟลเดอร์  
/home/pi/coding\_train/images จากนั้นจัดชื่อไฟล์รูปภาพทุกชิ้น



ตัวอย่างเช่น รูปภาพ Microbit = image0, สายแพร = image 6, Raspberry Pi = image12 และ  
Maqueen = image17

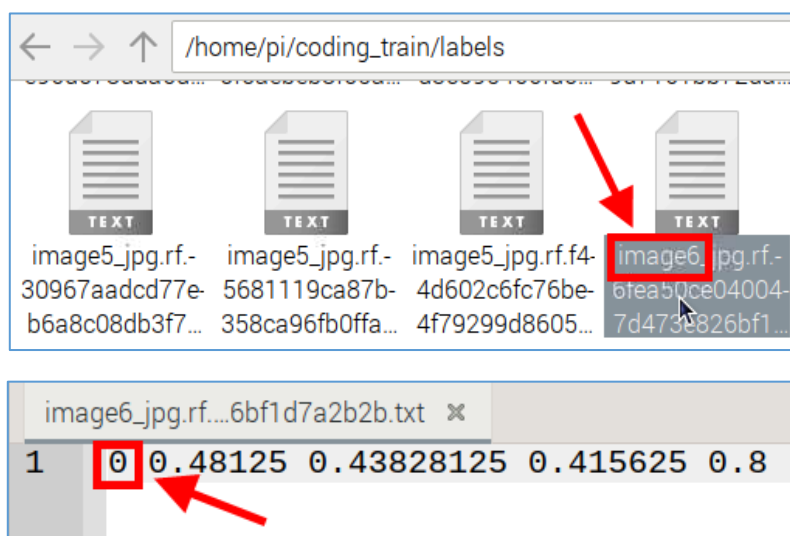
4.31 เข้าไฟล์เดอร์ /home/pi/coding\_train/labels จากนั้นเปิดไฟล์ลาเบล .txt ตามชื่อไฟล์รูปภาพที่ได้  
ในข้อ 4.30 ตัวอย่างเช่น

- รูปภาพ Microbit = image0



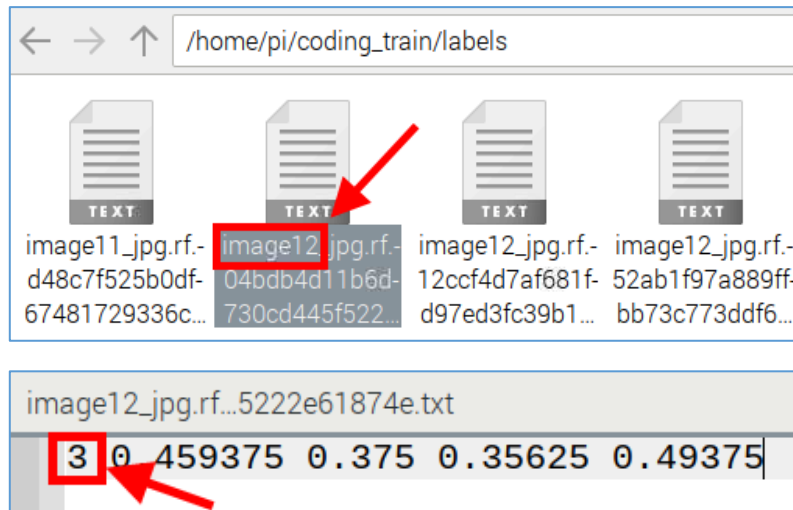
จะเห็นว่า id ของ Microbit คือ 2

- รูปภาพสายแพร์ = image 6



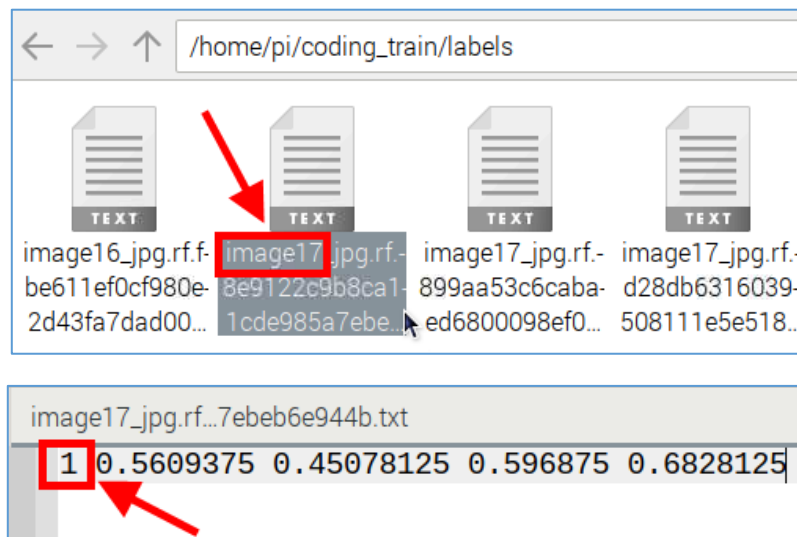
จะเห็นว่า id ของสายแพร์ คือ 0

- รูปภาพ Raspberry Pi = image12



จะได้ว่า id ของ Raspberry Pi คือ 3

- รูปภาพ Maqueen = image17



จะได้ว่า id ของ Maqueen คือ 1

สรุปว่า id = 0 คือ สายแพร์

id = 1 คือ Maqueen

id = 2 คือ Microbit

id = 3 คือ Raspberry Pi

หมายเหตุ : แต่ละคนจะได้ id ต่างกันออกไป

#### 4.32 พิมพ์โค้ดคำสั่งใน coding\_train.yaml ดังรูป

train: ตำแหน่งของโฟลเดอร์ Dataset ที่อยู่ใน Google Drive

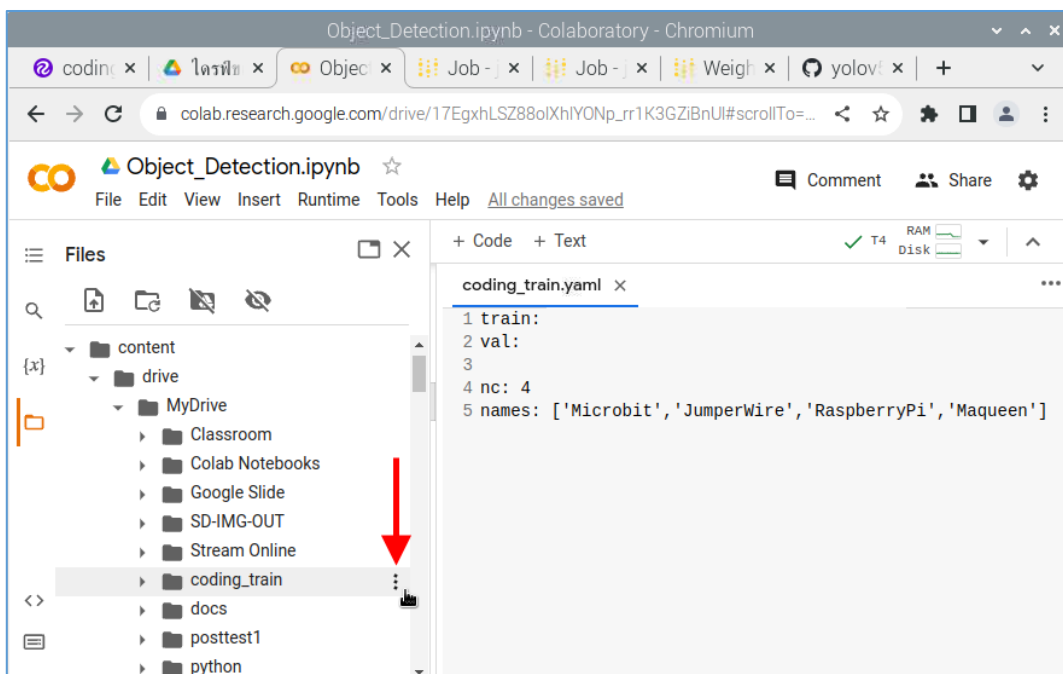
val: ตำแหน่งของโฟลเดอร์ Dataset ที่อยู่ใน Google Drive

nc: จำนวนวัตถุที่ทำการตรวจจับ

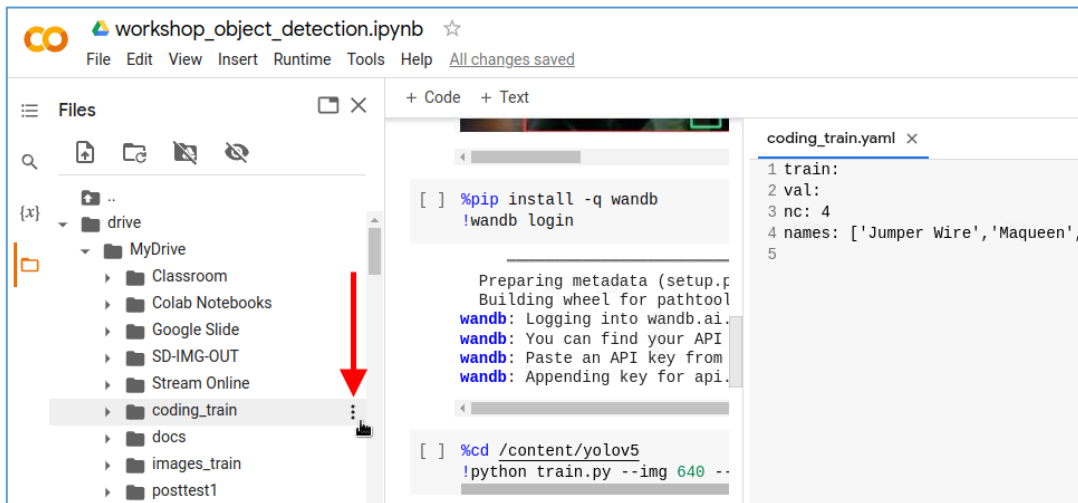
names: ['ชื่อวัตถุที่มี id = 0', 'ชื่อวัตถุที่มี id = 1', 'ชื่อวัตถุที่มี id = 2', 'ชื่อวัตถุที่มี id = 3']

```
coding_train.yaml ×
1 train:
2 val:
3 nc: 4
4 names: ['Jumper Wire', 'Maqueen', 'Microbit', 'Raspberry Pi']
5 |
```

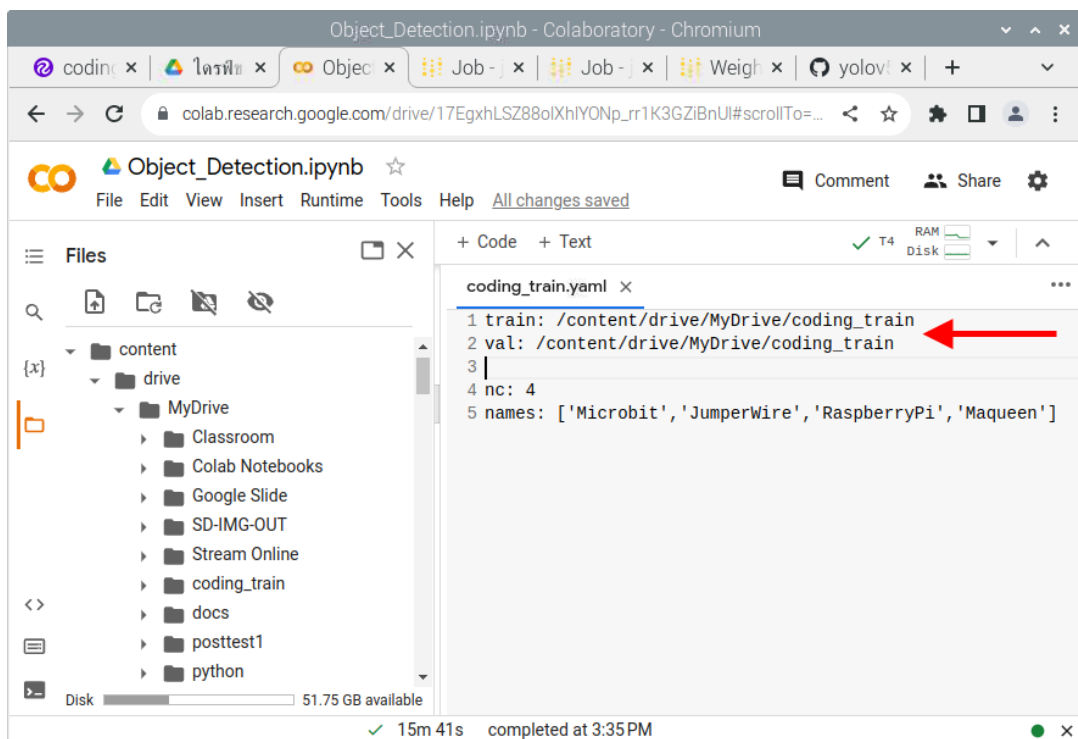
#### 4.33 คลิกที่จุด 3 จุดด้านล่างโฟลเดอร์ coding\_train (อยู่ในโฟลเดอร์ drive/MyDrive)



#### 4.34 คลิก Copy path เพื่อคัดลอกตำแหน่งของชุดข้อมูล Dataset ที่อยู่ใน Google Drive



#### 4.35 นำ Path ของโฟลเดอร์ coding\_train มาวางด้านหลังโค้ดคำสั่ง train: และ val: ดังรูป





4.36 เปิดไฟล์ yolov5s.yaml ซึ่งอยู่ในโฟลเดอร์ models มาแก้ไขโค้ดคำสั่ง nc: 80 ให้เป็น **nc: จำนวนวัตถุที่จะตรวจจับ** เหมือนไฟล์ coding\_train.yaml

```
*yolov5s.yaml x
1 # YOLOv5  by Ultralytics, AGPL-3.0 license
2
3 # Parameters
4 nc: 4 # number of classes
5 depth_multiple: 0.33 # model depth multiple
6 width_multiple: 0.50 # layer channel multiple
7 anchors:
8 - [10,13, 16,30, 33,23] # P3/8
9 - [30,61, 62,45, 59,119] # P4/16
10 - [116,90, 156,198, 373,326] # P5/32
11
```

4.37 รันไฟล์ train.py ซึ่งเป็นโค้ดสำหรับการ Train โดยคลิก **+ Code** แล้วพิมพ์คำสั่งดังรูป

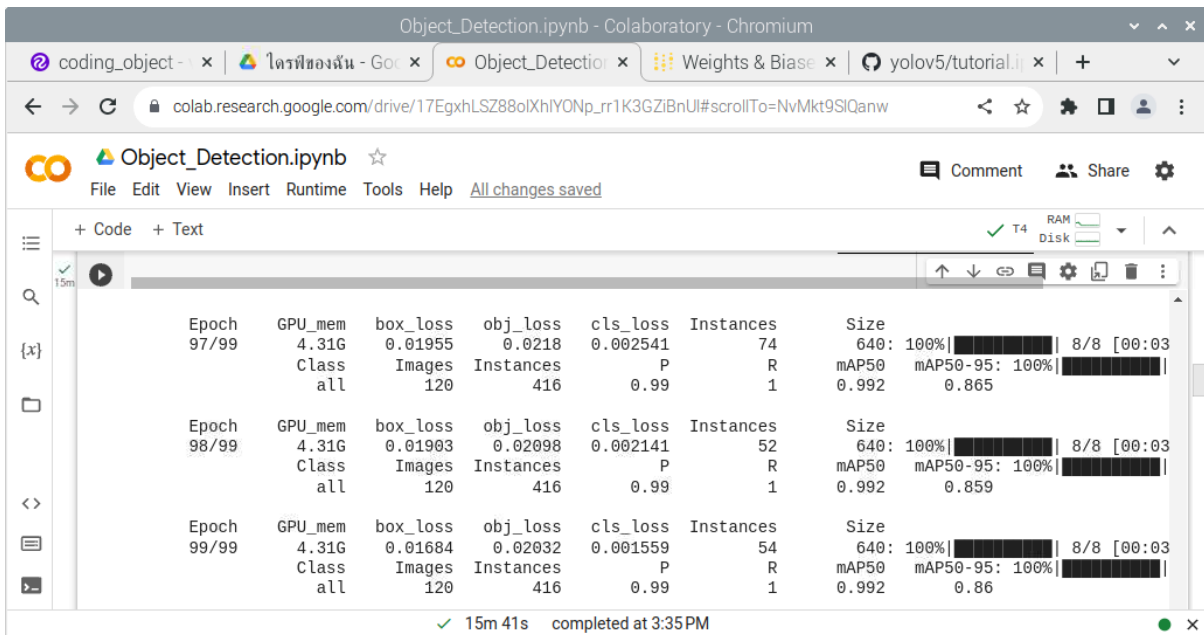
```
Object_Detection.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
%cd /content/yolov5
!python train.py --img 640 --epochs 100 --data coding_train.yaml --cfg ./models/yolov5s.yaml --weights 'yolov5s.pt'
```

```
/content/yolov5
wandb: WARNING Δ wandb is deprecated and will be removed in a future release. See supported integrations at https://g
wandb: Currently logged in as: vivy284. Use `wandb login --relogin` to force relogin
train: weights=yolov5s.pt, cfg=./models/yolov5s.yaml, data=coding_train.yaml, hyp=data/hyps/hyp.scratch-low.yaml, epo
github: up to date with https://github.com/ultralytics/yolov5 ✓
YOLOv5  by v7.0-218-g9e97ac3 Python-3.10.12 torch-2.0.1+cu118 CUDA:0 (Tesla T4, 15102MiB)

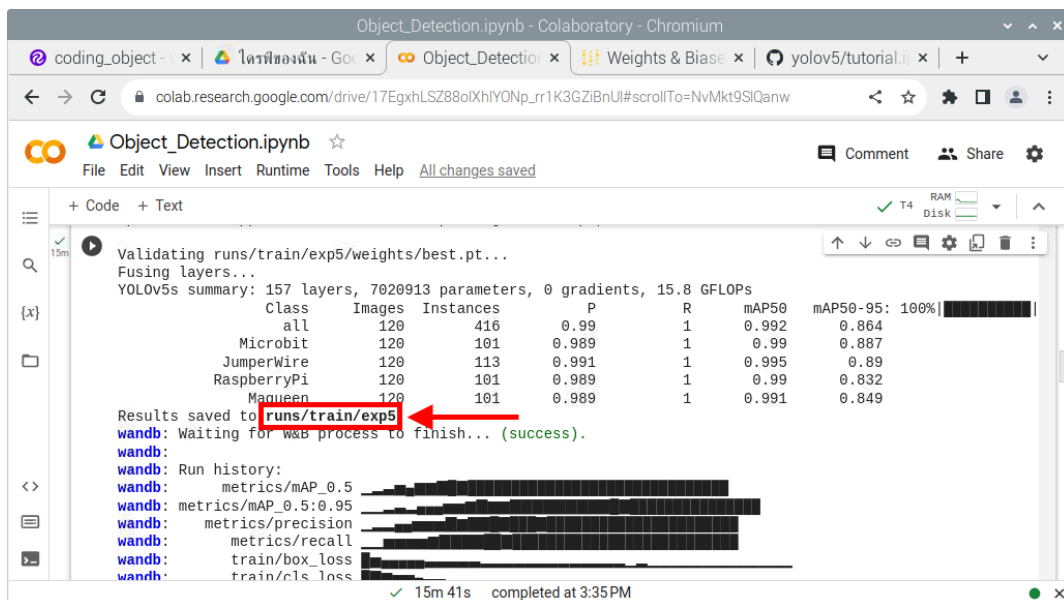
hyperparameters: lr0=0.01, lrf=0.01, momentum=0.937, weight_decay=0.0005, warmup_epochs=3.0, warmup_momentum=0.8, warr
TensorBoard: Start with 'tensorboard --logdir runs/train', view at http://localhost:6006/
wandb: Tracking run with wandb version 0.15.10
wandb: Run data is saved locally in /content/yolov5/wandb/run-20230918_081949-c1qxzgy8
wandb: Run `wandb offline` to turn off syncing.
wandb: Syncing run revived-meadow-5
wandb: * View project at https://wandb.ai/vivy284/YOLOv5
wandb:  View run at https://wandb.ai/vivy284/YOLOv5/runs/c1qxzgy8
COMET WARNING: Comet credentials have not been set. Comet will default to offline logging. Please set your credential
✓ 15m 41s completed at 3:35PM
```

**หมายเหตุ:** (--mg 640 คือ ขนาดภาพ, --epochs 100 คือ จำนวนครั้งในการ Train, --data coding\_train.yaml คือ ไฟล์ Dataset, --cfg ./models/yolov5s.yaml คือ ไฟล์ config, --weights 'yolov5s.pt' คือ โมเดลที่ใช้ Train)

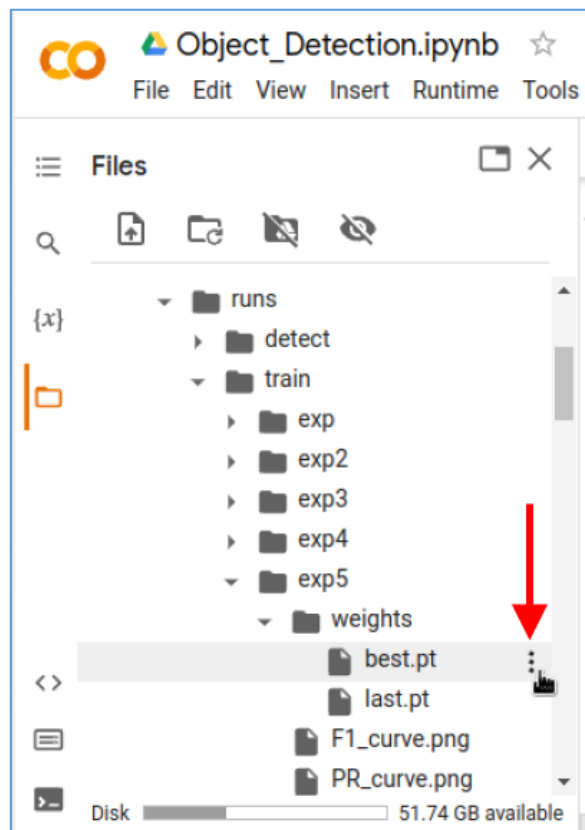
#### 4.38 ระบบจะทำการ Train จำนวน 100 รอบ รอจนการ Train เสร็จสมบูรณ์



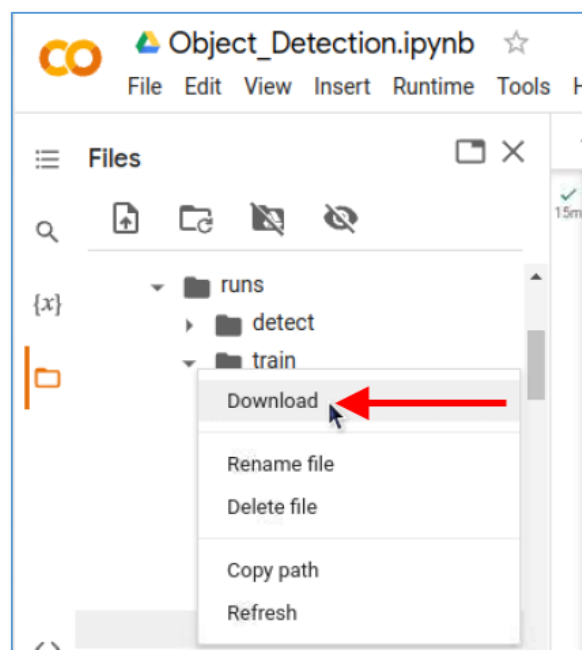
#### 4.39 ผลลัพธ์ที่ได้จะอยู่ในโฟลเดอร์ runs/train/exp



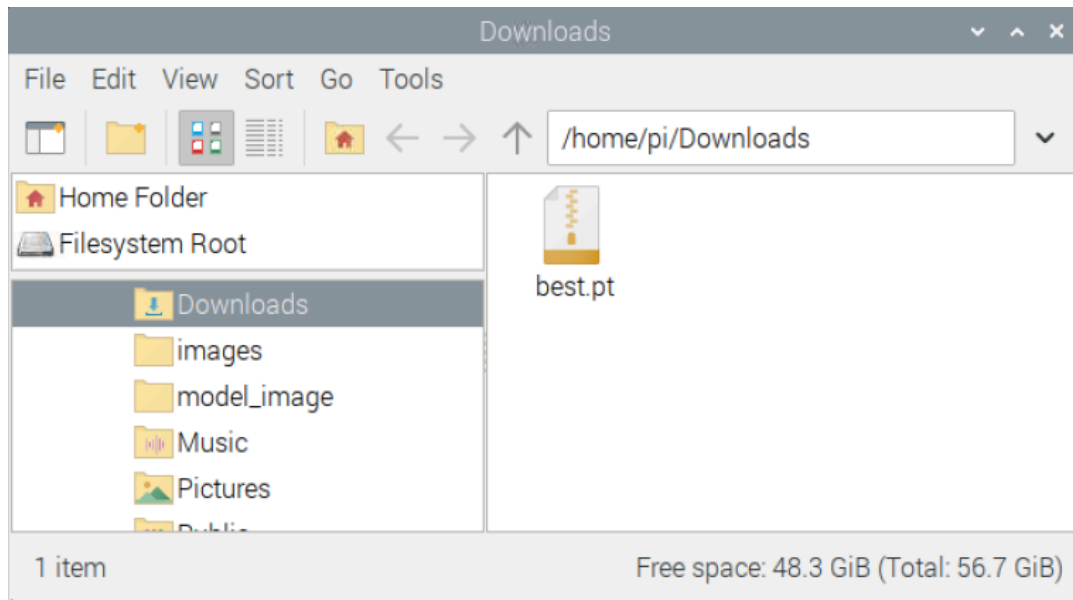
4.40 คำนวณโหลดไฟล์ที่ทำการ Train สำเร็จ โดยคลิกไปที่โฟลเดอร์ runs/train/exp/weights จากนั้นคลิกจุด 3 จุดที่ไฟล์ best.pt



4.41 คลิก Download



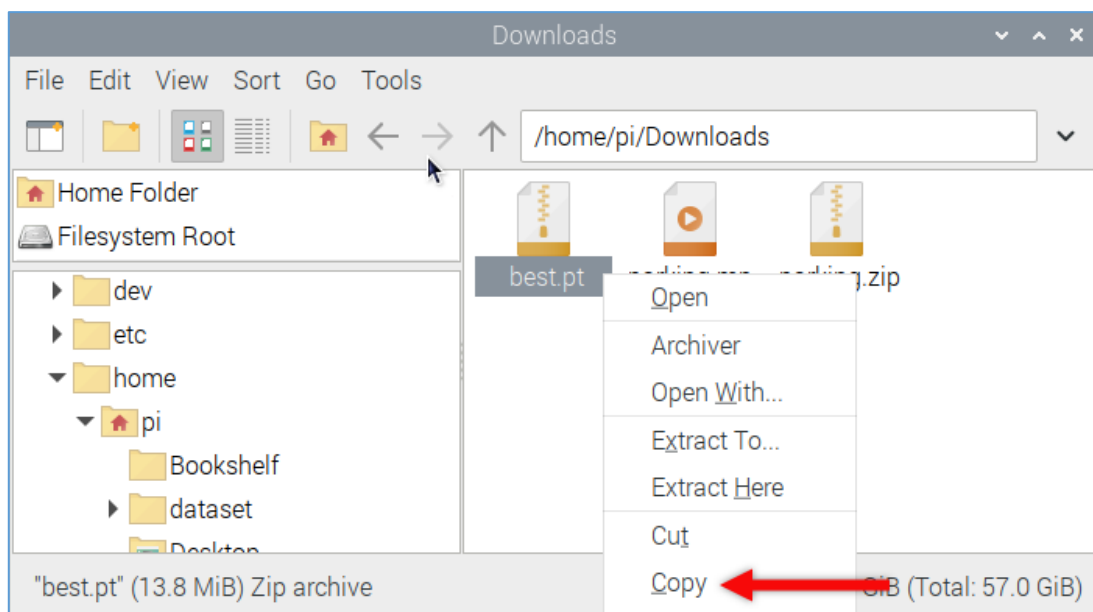
4.42 ไฟล์จะถูกดาวน์โหลดไว้ในโฟลเดอร์ /home/pi/Downloads



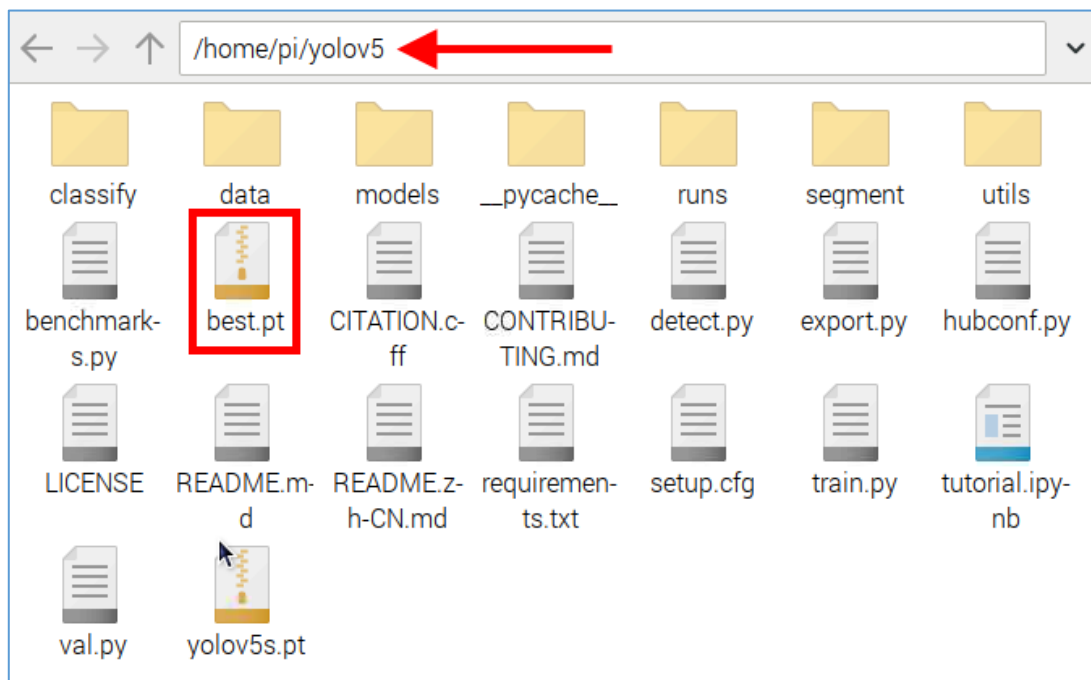
## 5. นำโมเดลที่ผ่านการ Train มาใช้งาน

5.1 ไฟล์ที่ผ่านการ Train (best.pt) จะอยู่ในโฟลเดอร์ Downloads ให้คลิกขวาที่ไฟล์แล้วเลือกคำสั่ง

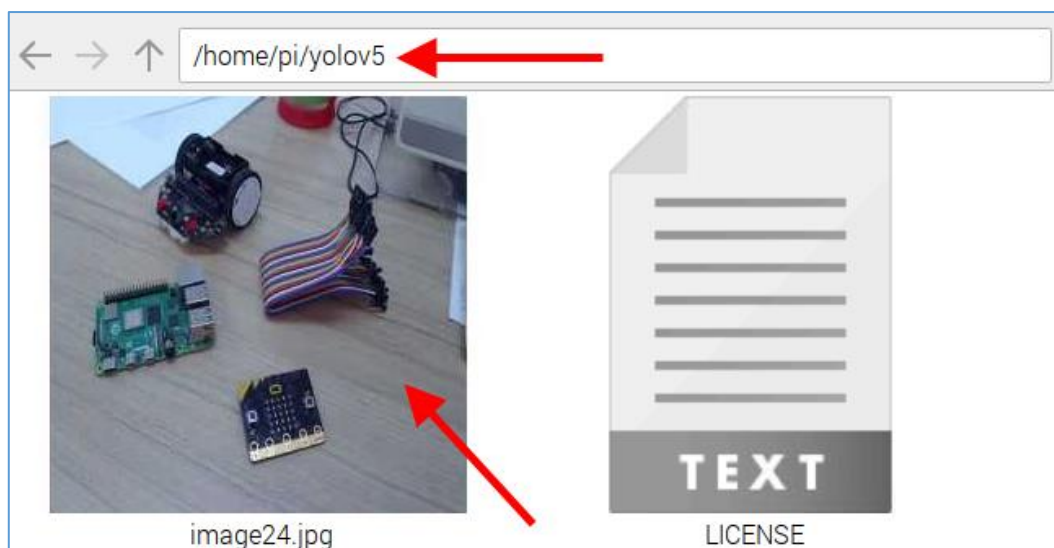
Copy



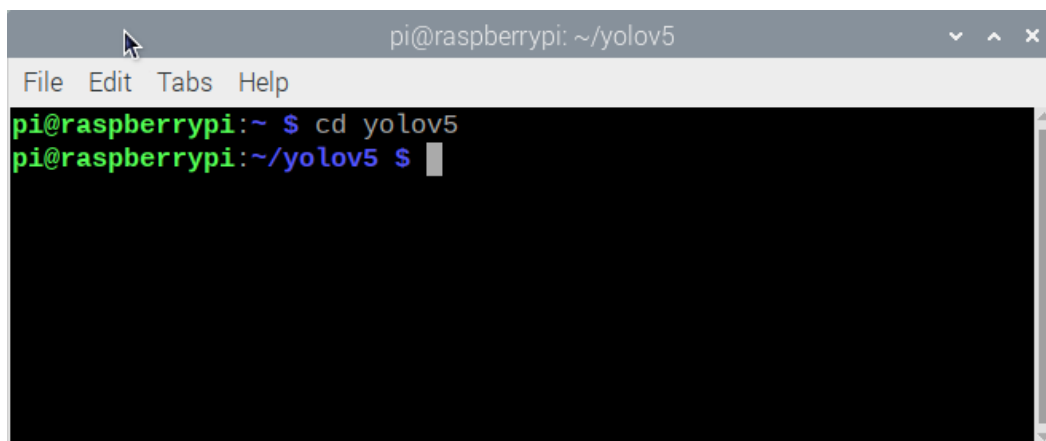
5.2 คัดลอกไฟล์ best.pt มาไว้ในโฟลเดอร์ /home/pi/yolov5



5.3 ทดสอบการตรวจจับวัตถุ โดยคัดลอกรูปภาพในโฟลเดอร์ images มาไว้ในโฟลเดอร์ yolov5 ยกตัวอย่างไฟล์ image24.jpg



5.4 ทดลอง Classify Object ด้วยรูปภาพ โดยเปิด Terminal แล้วเข้าไปในโฟลเดอร์ yolov5 โดยพิมพ์คำสั่ง `cd yolov5`

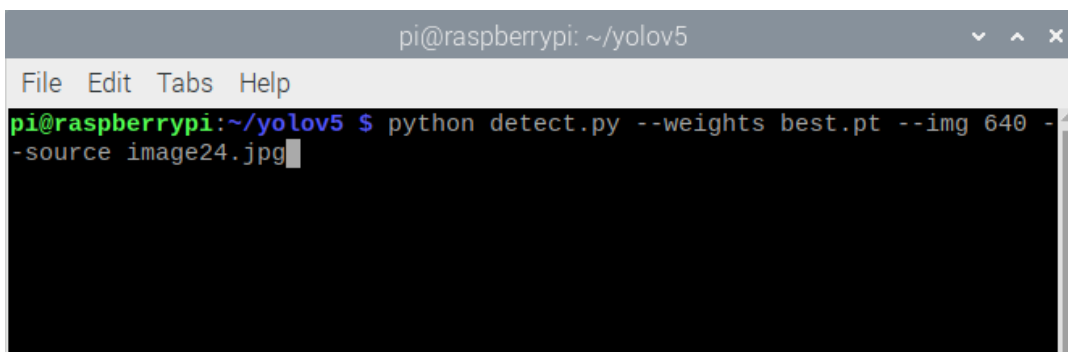


```

pi@raspberrypi: ~/yolov5
File Edit Tabs Help
pi@raspberrypi:~ $ cd yolov5
pi@raspberrypi:~/yolov5 $

```

5.5 รันไฟล์ detect.py โดยพิมพ์คำสั่ง `python detect.py --weights best.pt --img 640 --source image24.jpg`

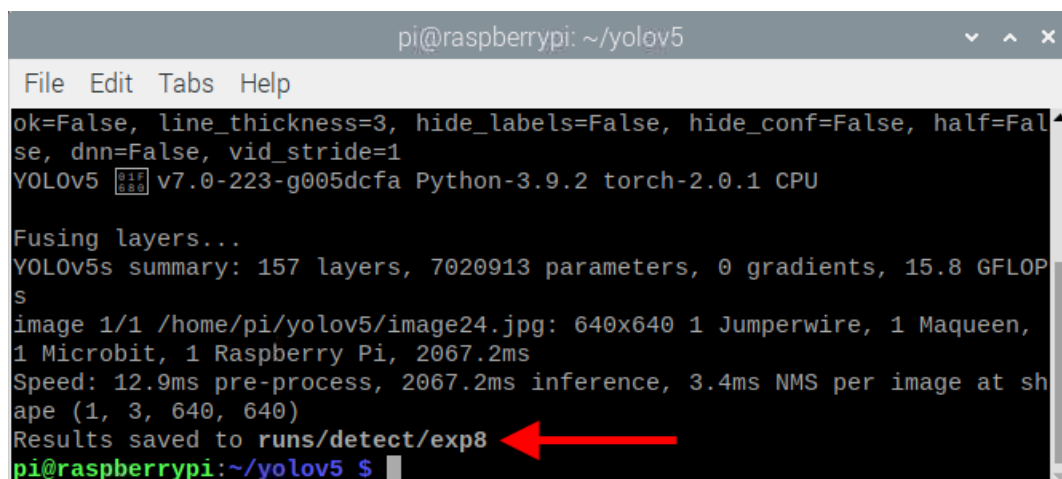


```

pi@raspberrypi: ~/yolov5
File Edit Tabs Help
pi@raspberrypi:~/yolov5 $ python detect.py --weights best.pt --img 640 --source image24.jpg

```

5.6 ผลลัพธ์จะถูกบันทึกไว้ในโฟลเดอร์ `/home/pi/yolov5/runs/detect/exp` ดังรูป



```

pi@raspberrypi: ~/yolov5
File Edit Tabs Help
ok=False, line_thickness=3, hide_labels=False, hide_conf=False, half=False,
dnn=False, vid_stride=1
YOLOv5 v7.0-223-g005dcfa Python-3.9.2 torch-2.0.1 CPU
Fusing layers...
YOLOv5s summary: 157 layers, 7020913 parameters, 0 gradients, 15.8 GFLOPs
image 1/1 /home/pi/yolov5/image24.jpg: 640x640 1 Jumperwire, 1 Maqueen,
1 Microbit, 1 Raspberry Pi, 2067.2ms
Speed: 12.9ms pre-process, 2067.2ms inference, 3.4ms NMS per image at shape (1, 3, 640, 640)
Results saved to runs/detect/exp8
pi@raspberrypi:~/yolov5 $

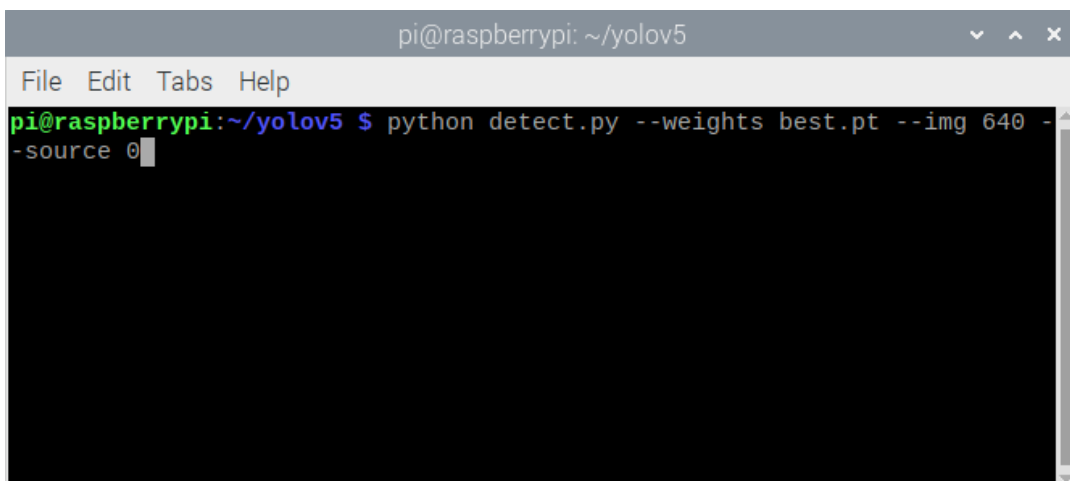
```

5.7 เข้าไปในโฟลเดอร์ /home/pi/yolov5/runs/detect/exp จะปรากฏรูปภาพผลลัพธ์ ดังรูป



5.8 ทดลองรัน โปรแกรมตรวจจับวัตถุ Object Detection กับกล้อง Webcam โดยพิมพ์คำสั่ง

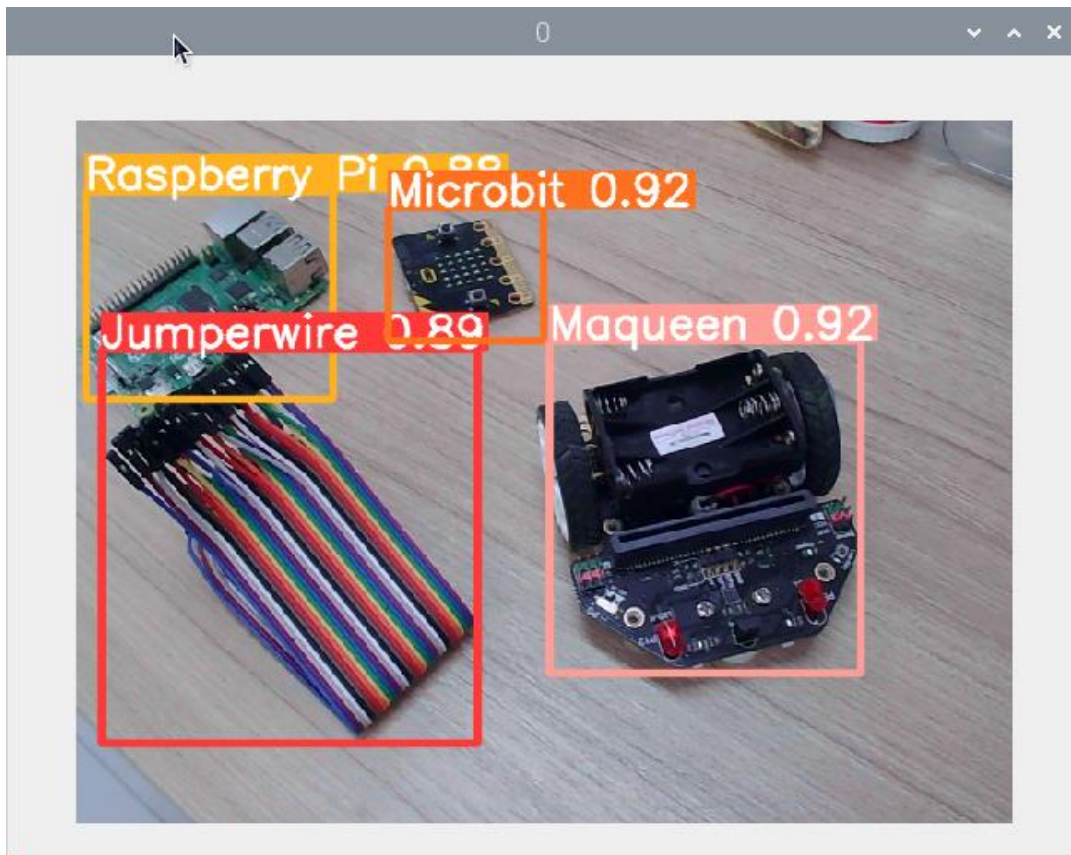
```
python detect.py --weights best.pt --img 640 --source 0
```



```
pi@raspberrypi: ~/yolov5
File Edit Tabs Help
pi@raspberrypi:~/yolov5 $ python detect.py --weights best.pt --img 640 --source 0
```



5.9 จะปรากฏหน้าจอกล๊องขึ้นมา ให้นำวัตถุไปส่องด้านหน้ากล้องเพื่อตรวจจับวัตถุ



5.10 เขียน โค้ดคำสั่งตรวจเช็คพื้นที่ว่างบนลานจอด

```

count_car_parking.py ✕
1 import cv2
2 import torch
3 import numpy as np
4 import time
5
6 points=[]
7 def POINTS(event, x, y, flags, param):
8     if event == cv2.EVENT_MOUSEMOVE:
9         colorsBGR = [x, y]
10        print(colorsBGR)
11 cv2.namedWindow('FRAME')
12 cv2.setMouseCallback('FRAME', POINTS)
13
14 model = torch.hub.load('ultralytics/yolov5', 'yolov5s', pretrained=True)
15 cap = cv2.VideoCapture('parking.mp4')
16
17 area = [(4, 234), (4, 310), (480, 255), (426, 202)]
18 while True:
19     ret, frame = cap.read()
20     if not ret:
21         break
22     frame = cv2.resize(frame, (640, 360))
23
24     results = model(frame)
25     list = []
26     for index, row in results.pandas().xyxy[0].iterrows():
27         x1 = int(row['xmin'])
28         y1 = int(row['ymin'])
29         x2 = int(row['xmax'])
30         y2 = int(row['ymax'])
31         d = (row['name'])
32         cx = int(x1 + x2) // 2
33         cy = int(y1 + y2) // 2
34
35         if 'car' in d:
36             results = cv2.pointPolygonTest(np.array(area, np.int32), ((cx, cy)), False)
37             if results >= 0:
38                 #print(results)
39                 cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 0, 255), 3)
40                 cv2.putText(frame, str(d), (x1, y1), cv2.FONT_HERSHEY_PLAIN, 2, (255, 0, 0), 2)
41
42                 list.append([cx])
43     cv2.polylines(frame, [np.array(area, np.int32)], True, (0, 255, 0), 2)
44     a = (len(list))
45     print(a)
46     cv2.putText(frame, "Car = " + str(a), (50, 50), cv2.FONT_HERSHEY_PLAIN, 2, (255, 0, 0), 2)
47     cv2.putText(frame, "Empty Parking = " + str(12 - a), (50, 90), cv2.FONT_HERSHEY_PLAIN, 2, (0, 255, 0), 2)
48     cv2.imshow("FRAME", frame)
49     cv2.setMouseCallback("FRAME", POINTS)
50
51     #time.sleep(0.1)
52     if cv2.waitKey(1) & 0xFF == 27:
53         break
54     cap.release()
55     cap.destroyAllWindows()

```

หมายเหตุ : หากใช้โมเดลที่เทรนเอง จะต้องเปลี่ยนคำสั่งในบรรทัดที่ 14 เป็น

```
model = torch.hub.load('ultralytics/yolov5', 'custom', 'best.pt', force_reload=True)
```