

## ★ CiRA CORE คืออะไร

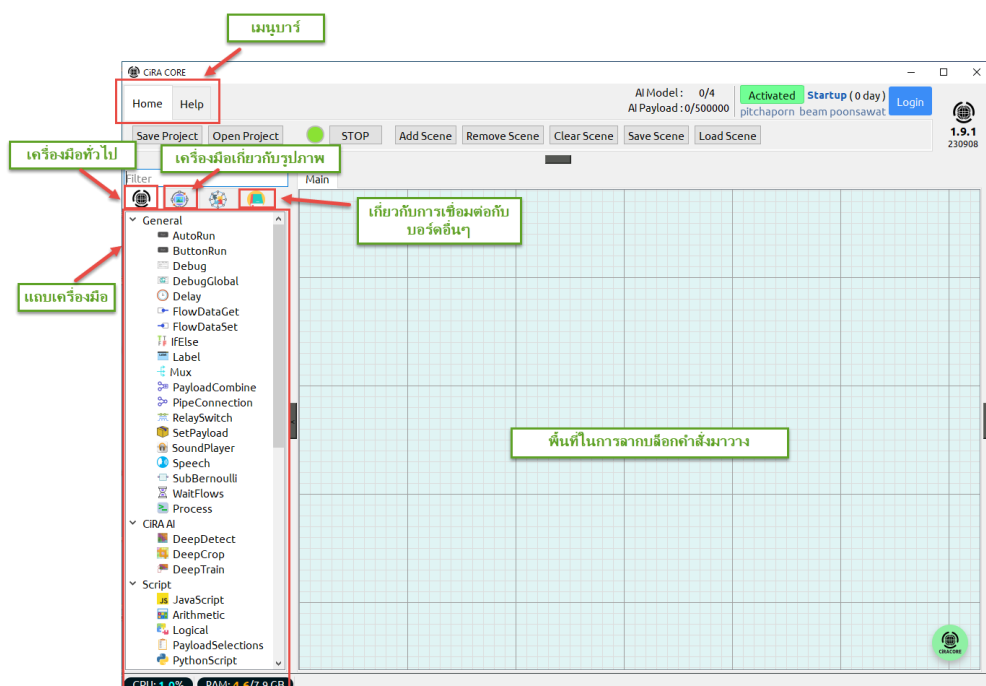
CiRA Core คือแพลตฟอร์มที่เป็น Core Technology ซึ่งถ้าเทียบกับเทคโนโลยีที่คล้ายๆ กัน เช่น ระบบปฏิบัติการแอนดรอยด์ที่สามารถเข้าไปอยู่ในสมาร์ทโฟน นาฬิกา กล้องดิจิทัล รวมถึงสมาร์ททีวี โดยเราสามารถสร้างอัลกอริทึมหรือแอปพลิเคชันต่างๆ เช่น Deep Learning ใส่เข้าไป เพื่อสร้างการเรียนรู้จดจำให้แก่ระบบ เช่น สร้างให้จดจำว่าของจดหมายรูปร่างหน้าตาเป็นแบบนี้ หลังจากการเรียนรู้จดจำแล้ว เมื่อนำไปใช้ในภาคอุตสาหกรรมก็ต้องมีการสั่งงาน อาทิ การสั่งหุ่นยนต์ให้หยิบของจดหมายออกจากสายพาน ดังนั้น CiRA Core คือแพลตฟอร์มกลางที่เชื่อมโยงแอปพลิเคชันต่างๆ ไปสู่การใช้งานจริง หรือก็คือตัวกลางระหว่างการเรียนรู้จดจำความคิดไปสู่การสั่งงาน

เป็น low-code platform ที่เรียนรู้วิธีใช้งานได้ง่าย ใช้งานง่ายด้วย block code ทำให้การเขียนโค้ดน้อยลง

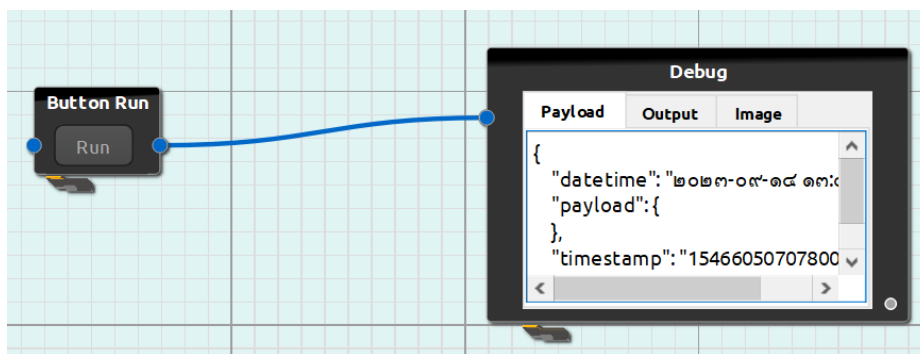
## ★ ทำไมต้อง CiRA CORE

- เชื่อมต่อกับอุปกรณ์ remote i/o ต่าง ๆ ได้เช่น Modbus/TCP, Arduino i/o, Serial Port, Robot เชื่อมต่อกับกล้อง IP camera, Webcam รวมถึงเปิดและบันทึกไฟล์ภาพ, ไฟล์วิดีโอได้
- เก็บข้อมูลและทำ label ได้ เก็บและสร้าง data set สำหรับการสร้างโมเดล AI ได้
- สามารถพัฒนาโมเดล AI และเลือกกระหว่างการใช้ CPU หรือ GPU สำหรับการฝึกอบรมให้มีความยืดหยุ่นและตัวเลือกประสิทธิภาพ
- ส่งข้อมูลต่อไปยังระบบอื่น ๆ ผ่านทาง MQTT หรือ REST API หรือสร้าง RTSP, MJPEG Server เพื่อส่ง Steam ได้
- สร้าง User Interface เบื้องต้น(ปุ่ม ภาพ ตัวหนังสือ LED)

## หน้าต่างโปรแกรม



## ★ เริ่มต้นในการเขียนโปรแกรม

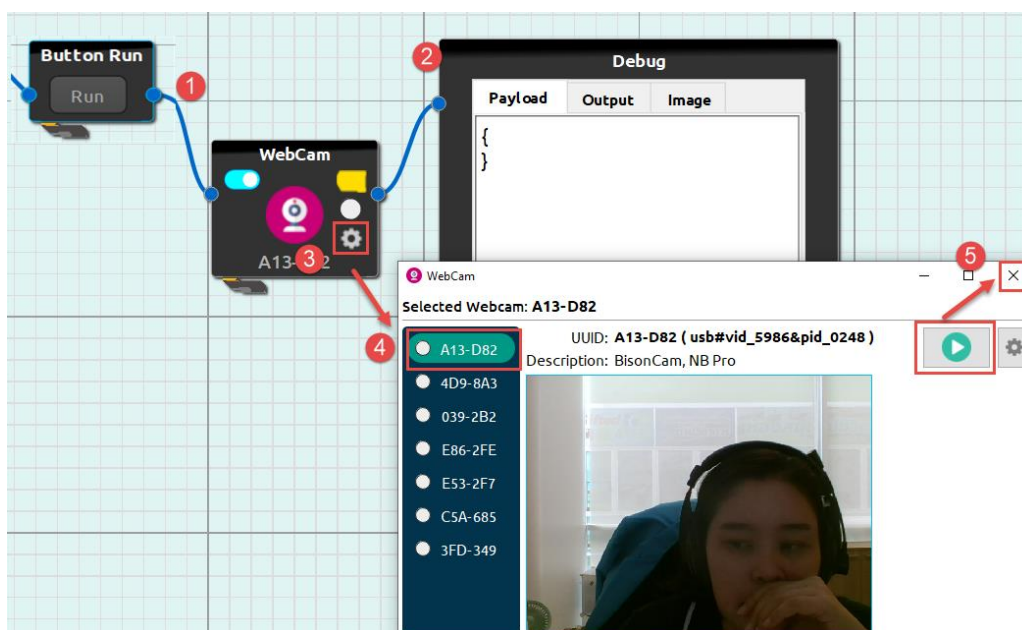


เป็นการทำงานแค่ 1 ครั้งในการกดปุ่ม Run

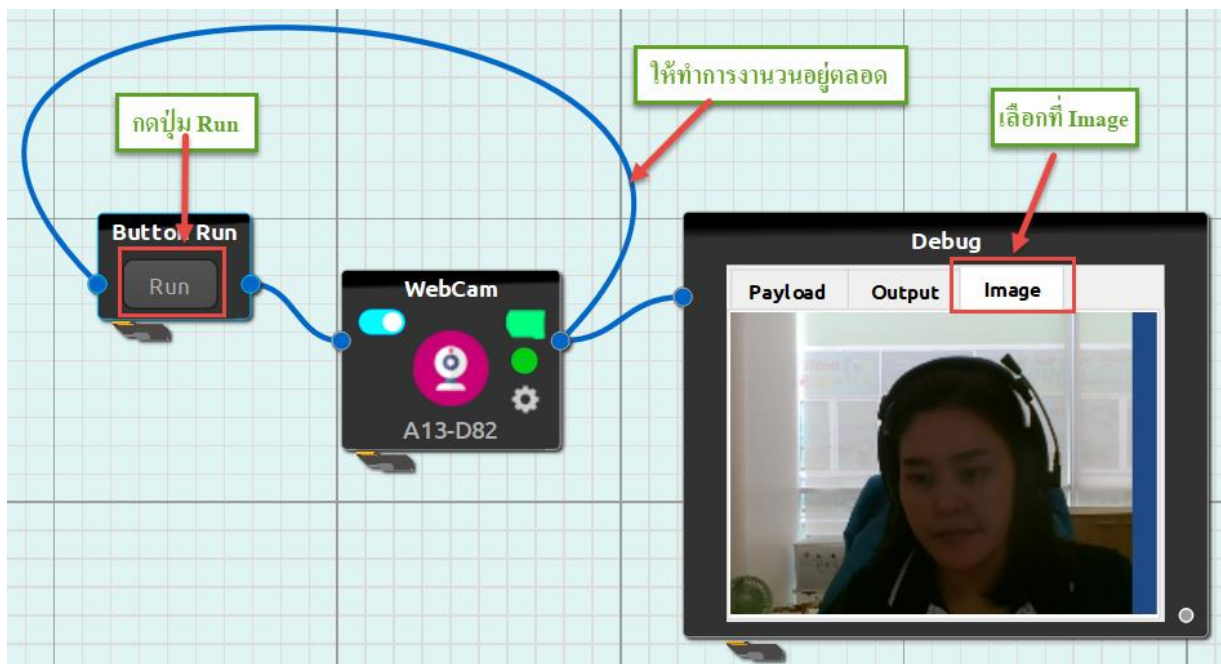
- Button Run เป็นการกดปุ่มเพื่อเริ่มต้นการรัน โปรแกรม
- Debug เป็นการแสดงผล มี
  - payload ในการดูข้อมูล
  - Output ผลลัพธ์ที่ส่งออกมา
  - Image แสดงรูปภาพ

## ★ ระบบตรวจจับวัตถุด้วย Deep Detect

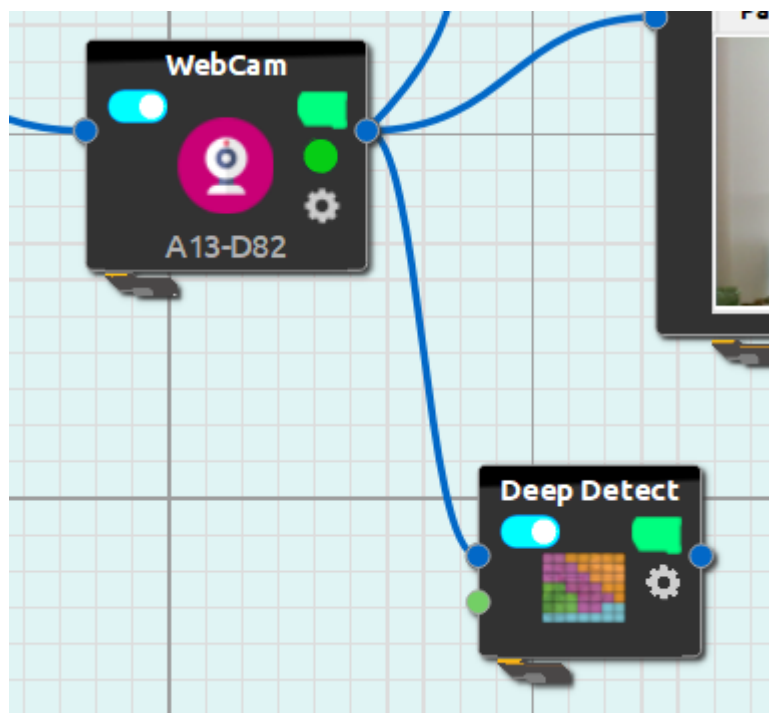
1. ใช้คำสั่ง Button Run และ WebCam ให้เชื่อมต่อสัญญาณกัน
2. ลากเครื่องมือ Debug และเชื่อมต่อสัญญาณ กับ WebCam
3. กดรูปการตั้งค่าในเครื่องมือ WebCam
4. เลือกกล้อง แล้วให้กดปุ่ม Play
5. แล้วก็ทำการปิดหน้าต่างไปได้เลย



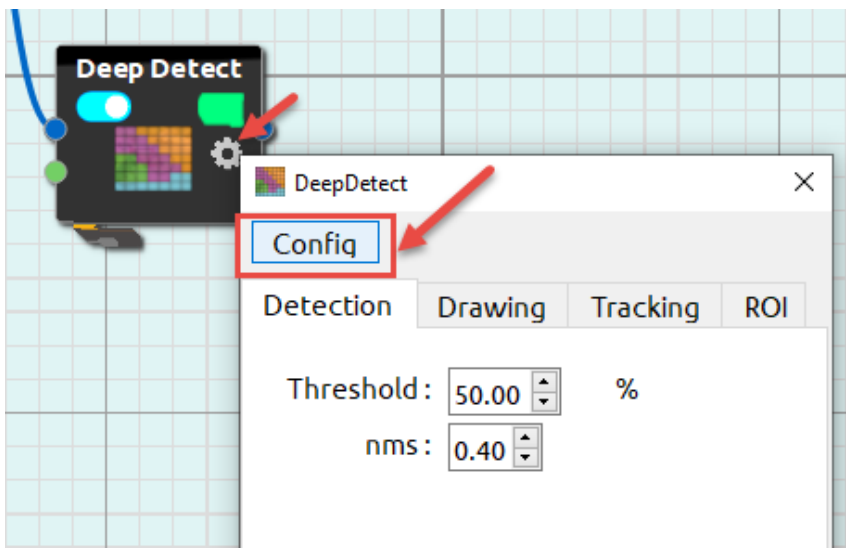
6. เลือกที่แท็บ Image ก็จะเปิดรูปภาพจากกล้องให้ และถ้าต้องการให้เป็นภาพต่อเนื่อง ก็ให้ลากเส้นความสัมพันธ์ จาก WebCam ย้อนกลับไปที่ปุ่ม Button Run



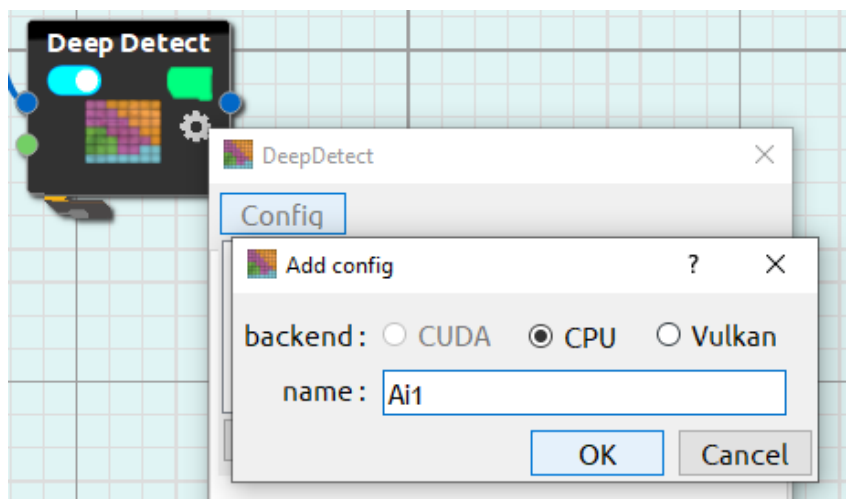
7. ทำการโหลด Model สำหรับ Platform CiRA CORE ที่ <https://blog.kls.ac.th/download> เลือก V3\_
8. ให้เลือกเครื่องมือ DeepDetect นำมาวางจากนั้นเชื่อมความสัมพันธ์



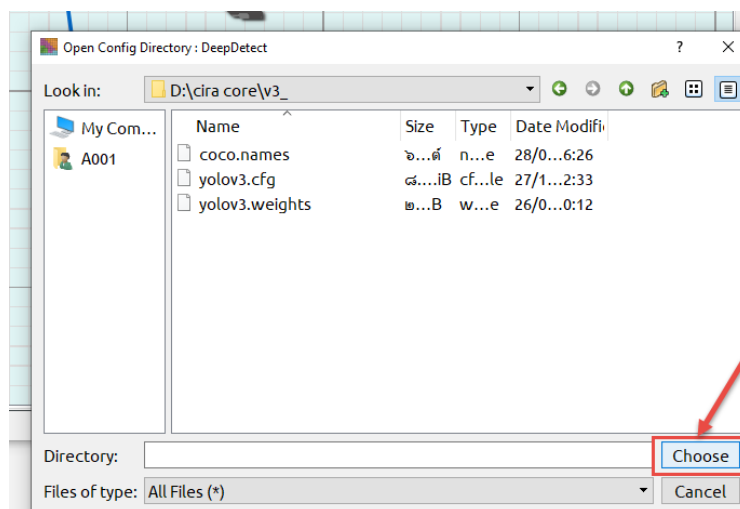
9. จากนั้นเลือกการตั้งค่าที่ Deep Detect แล้วกดที่ config



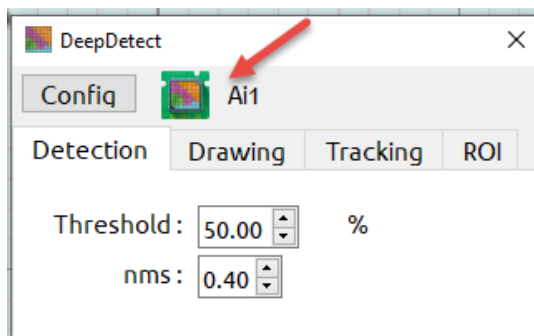
10. กด + ให้ตั้งชื่อ แล้วคลิกที่ปุ่ม OK โปรแกรมจะให้เลือกไฟล์ที่เราต้องการ config เลือก



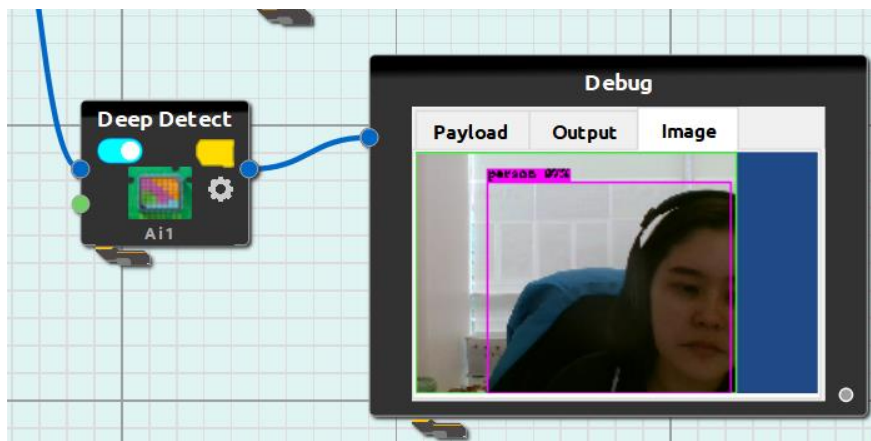
11. โปรแกรมจะให้เลือกสถานที่เก็บไฟล์ที่เราต้องการ config จากนั้นคลิกปุ่ม Choose



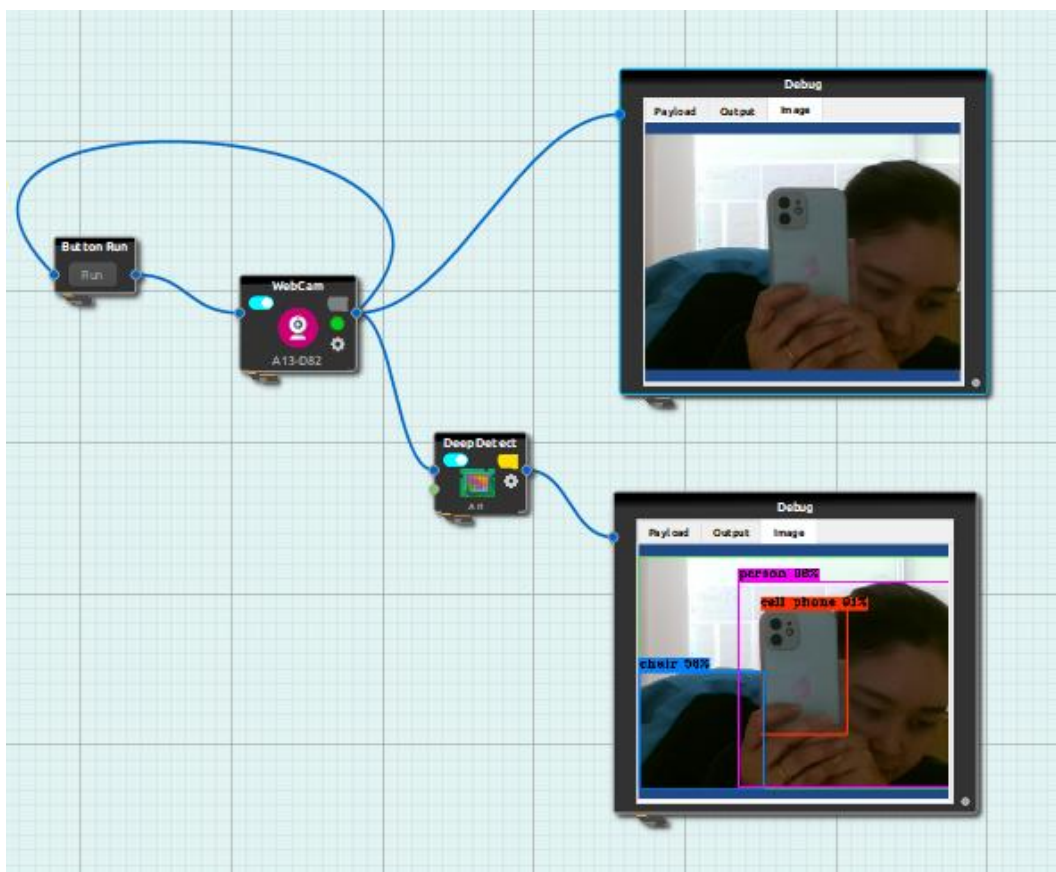
12. หน้าต่างที่ทำการ Config แล้ว จากนั้นกดปิดหน้าต่างโปรแกรม

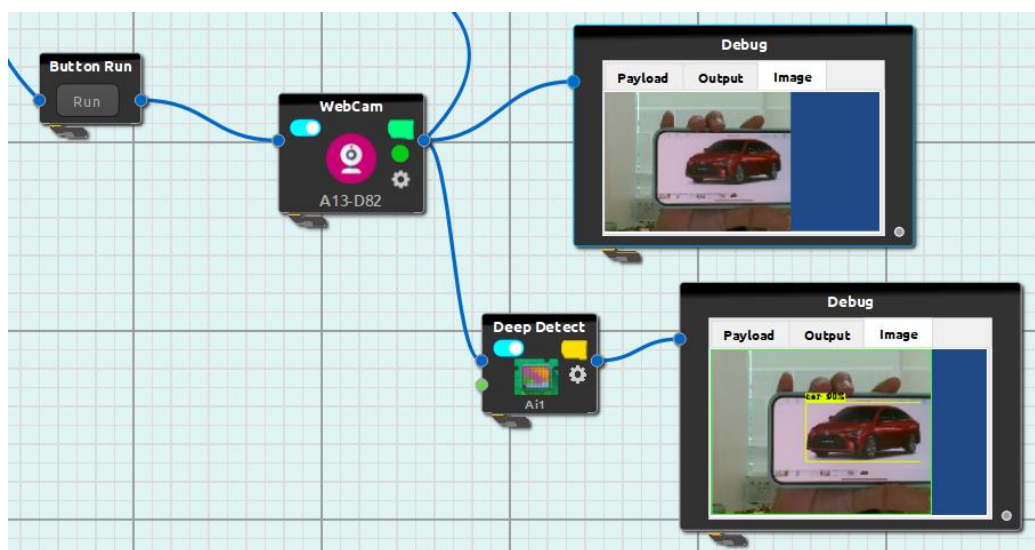
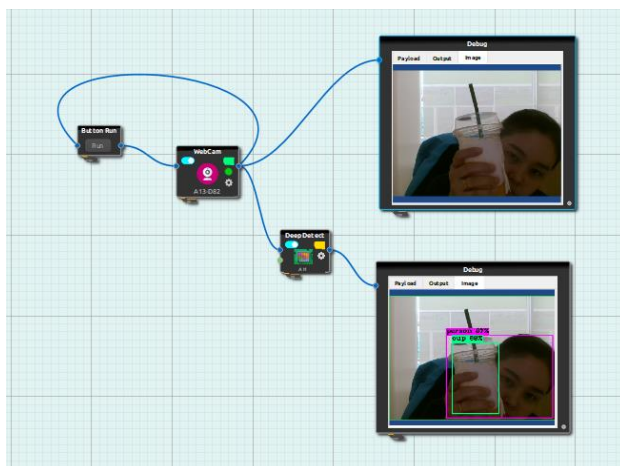


13. ให้เพิ่ม Debug อีก 1 ตัวเอาให้แสดงผล ให้เชื่อมต่อกับ DeepDetect



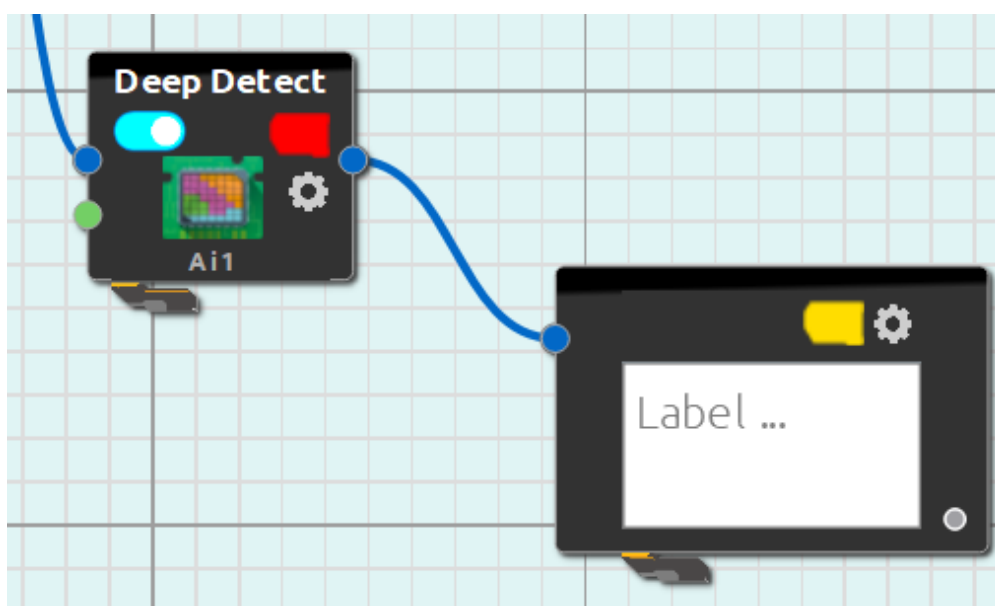
14. โปรแกรมเขาก็จะประมวลผล ว่าวัตถุที่เจอเป็นอะไร เช่น คน หรือ โทรศัพท์





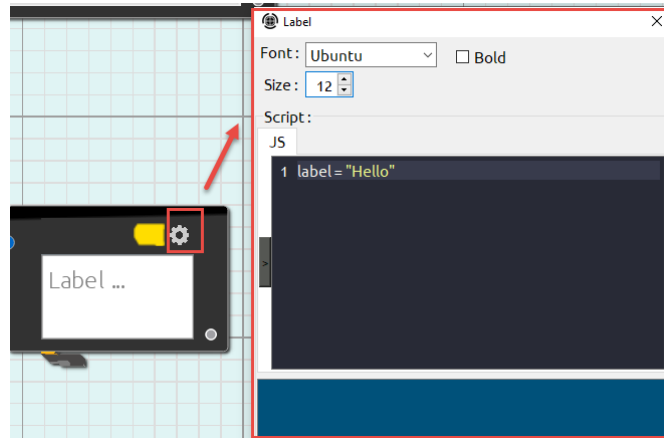
ตรวจสอบ และมีข้อความขึ้นที่ label

1. ทำเหมือนโปรเจ็คที่แล้ว จากนั้นให้ลาก label มาวางและเชื่อมความสัมพันธ์จาก DeepDetect ไปยัง Label





## 2. ทำการเขียนคำสั่งให้ Label โดยคลิกที่รูปพื้นเพื่อง



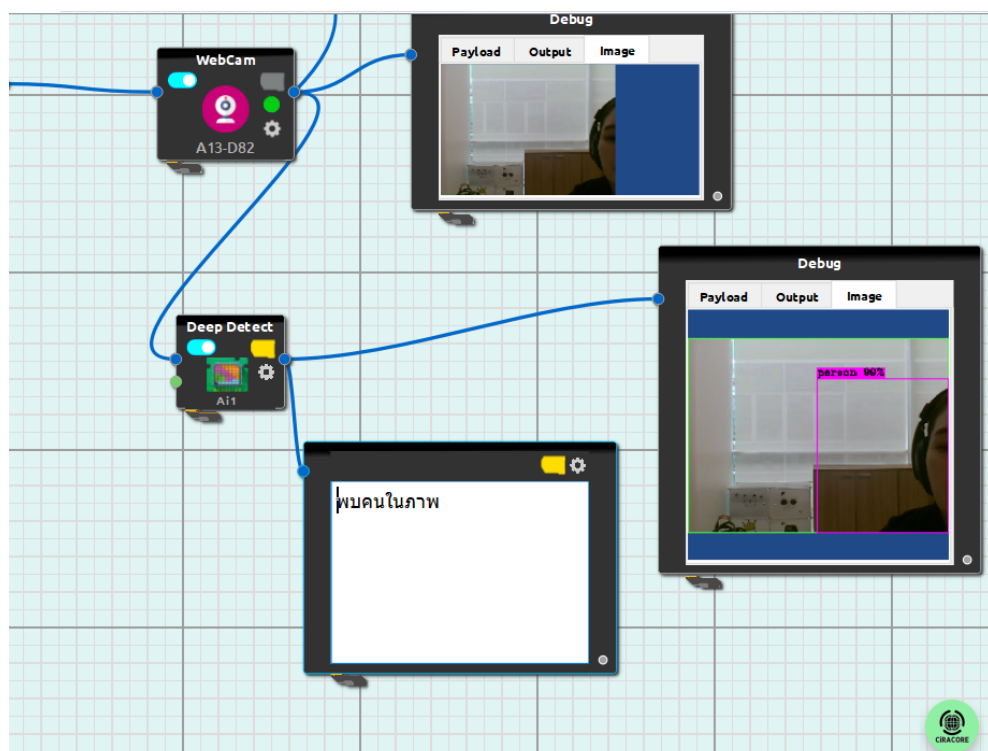
## 3. พิมพ์คำสั่งลง เสร็จแล้วกดปิด แล้วลองทดสอบโปรแกรม

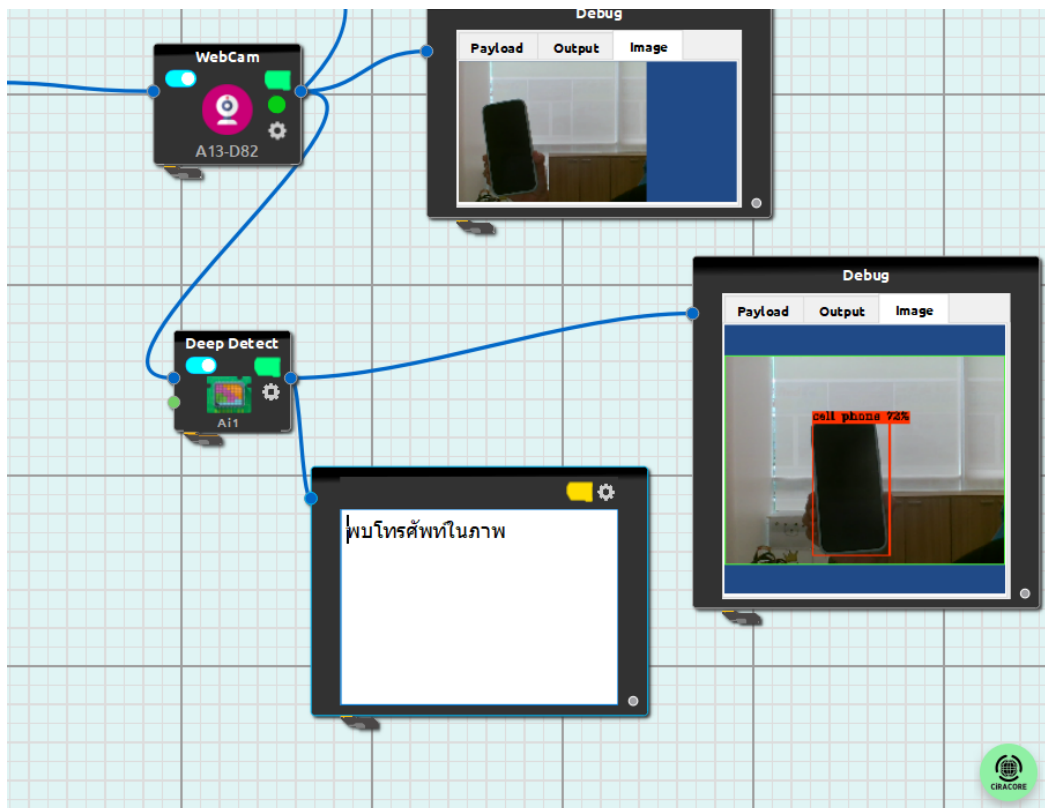
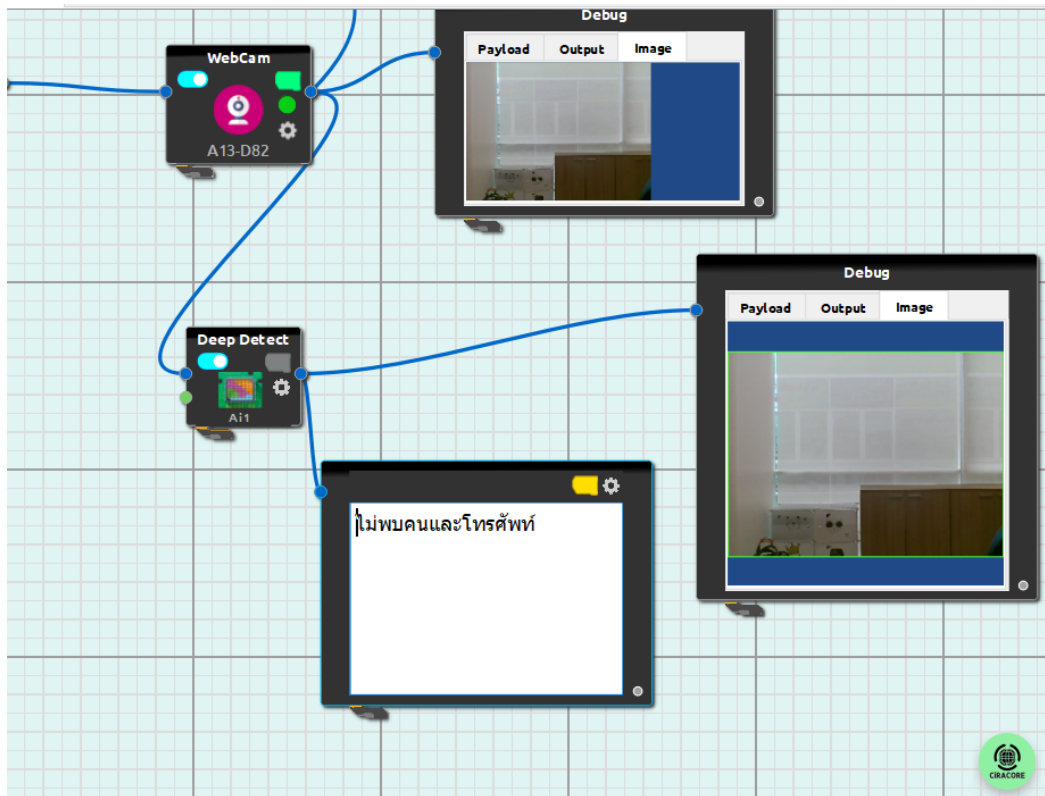
```

1 var objs=payload.DeepDetect.objects
2 var len = objs.length
3 label = "ไม่วัตถุและคน"
4 for (vari=0; i<len; i++){
5   if(objs[i].name == "person"){
6     label = "พบคนในภาพ"
7   }else if(objs[i].name == "cell phone"){
8     label = "พบโทรศัพท์ในภาพ"
9   }else if (objs[i].name == "car"){
10    label = "พบรถในภาพ"
11  }
12 }

```

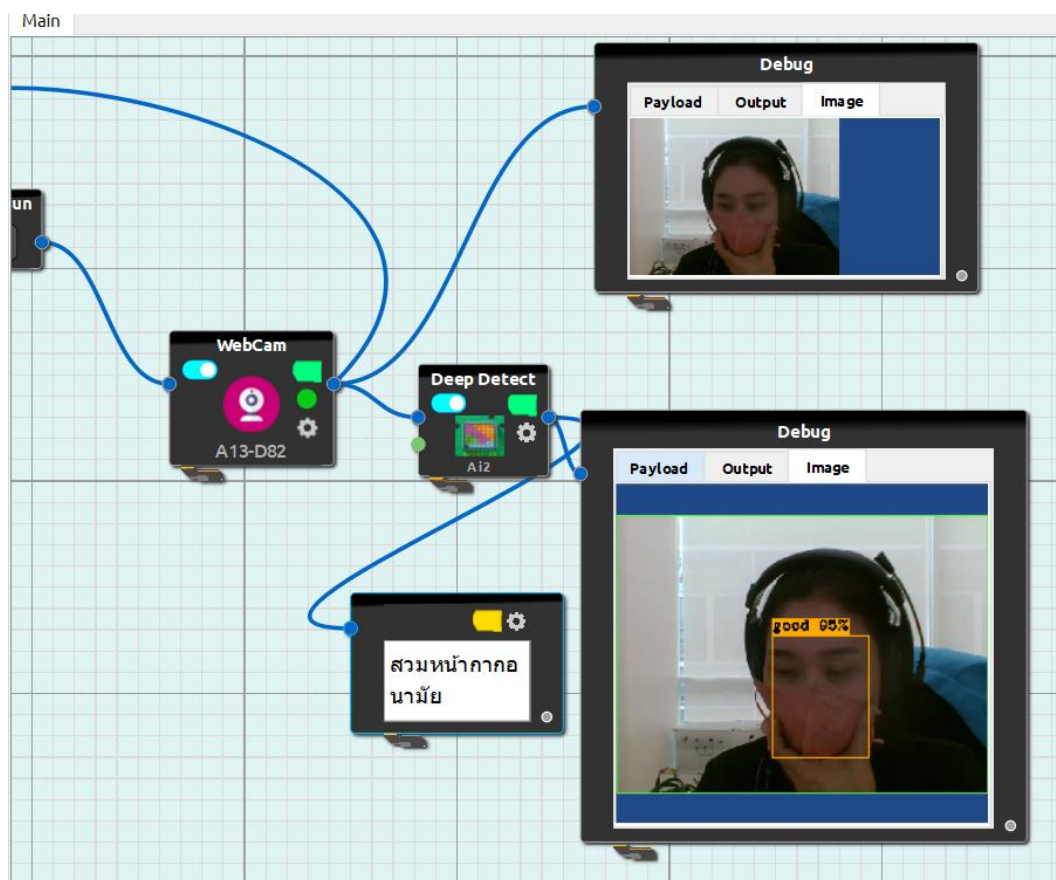
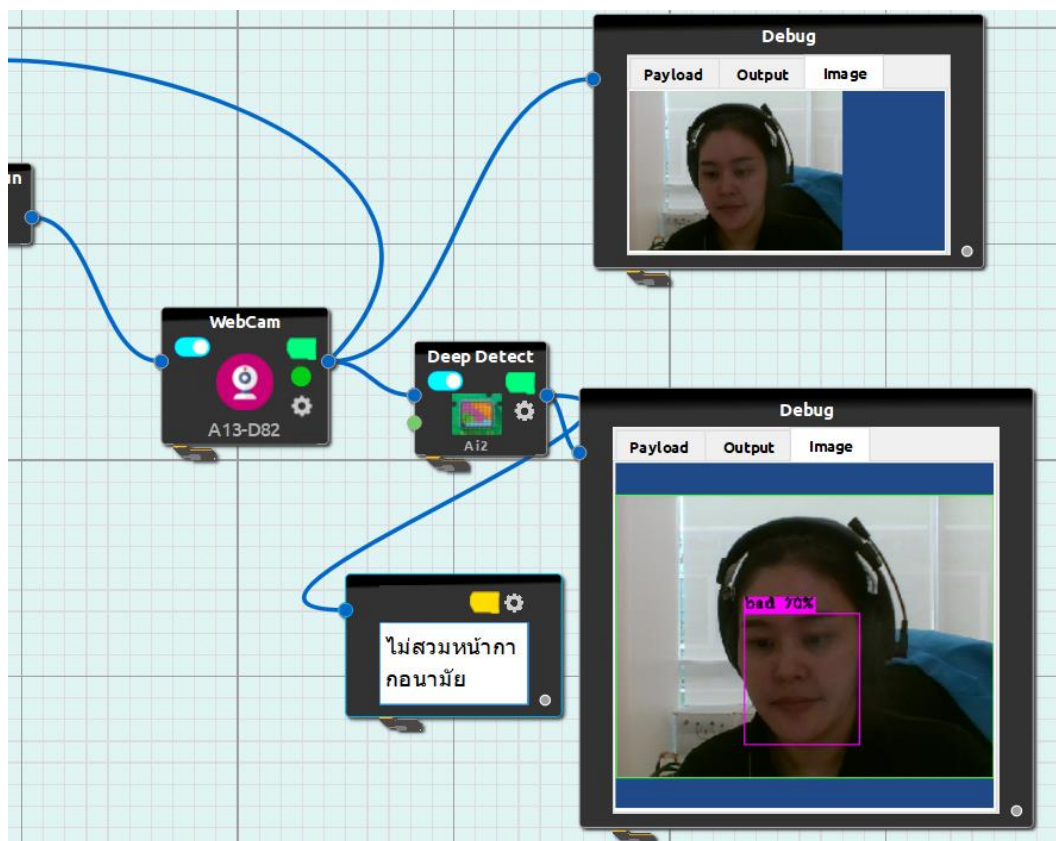
### ตัวอย่างการทดสอบ







## ★ ตรวจสอบการใส่หน้ากากอนามัย



```

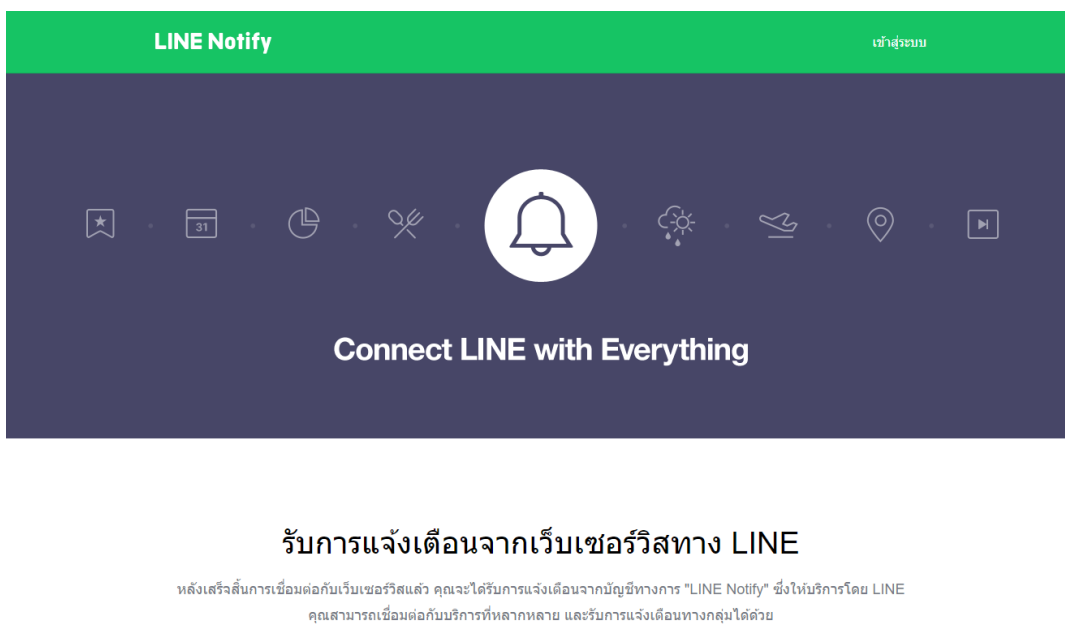
Label
Font: Ubuntu
Size: 12
Script:
JS
1 var objs = payload.DeepDetect.objects
2 var len =objs.length
3 label = "ไม่สามารถตรวจสอบได้"
4 for(var i=0; i<len; i++){
5   if(objs[i].name == "good"){
6     label = "สวมหน้ากากอนามัย"
7   }else if (objs[i].name == "bad"){
8     label = "ไม่สวมหน้ากากอนามัย"
9   }else{
10    label = "สวมหน้ากากอนามัยไม่ถูกต้อง"
11  }
12 }

```

## ★ การส่งข้อมูลไปยังทาง line ของเรา

### ขั้นตอนการขอ Token

การแจ้งเตือนผ่านทาง line จะต้องใช้ รหัส Token โดยเข้าไปที่ [notify-bot.line.me](https://notify-bot.line.me)



### 1. ทำการเพิ่มเพื่อน โดยการสแกน QR code



### 2. ตั้งค่า line ให้เชื่อมต่อกับ E-mail เมื่อทำทั้ง 2 ข้อแล้วให้คลิกเข้าสู่ระบบ

กรอก e-mail และ password ที่ผูกไว้กับ line

ได้รับรหัส เปิด line ในมือถือแล้วกรอกรหัสนี้

**LINE**

อีเมล  
kob13\_06@hotmail.com

รหัสผ่าน  
\*\*\*\*\*

**เข้าสู่ระบบ**

เข้าสู่ระบบด้วยวิธีอื่น

ใหม่ [เข้าสู่ระบบด้วยคิวอาร์โค้ด](#)

[ลืมอีเมลหรือรหัสผ่าน?](#)

**LINE**

**รหัสยืนยันตัวตน**

คุณจะต้องยืนยันตัวตนเพื่อการใช้งาน LINE อย่างปลอดภัย

**2777**

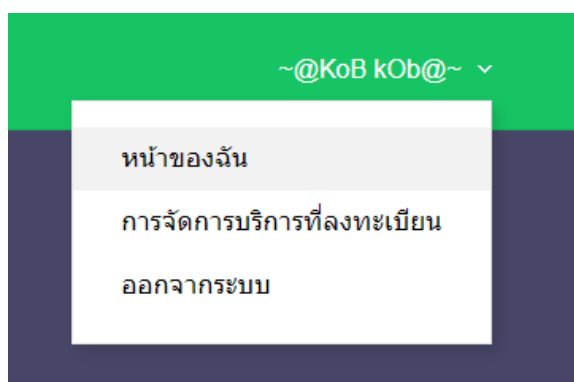
เหลือเวลา 02:57

โปรดใส่รหัสยืนยันตัวตนที่แอป LINE สำหรับ Android หรือ iOS

[เข้าสู่ระบบด้วยบัญชีอื่น](#)

ก็จะทำการเข้าสู่ระบบได้

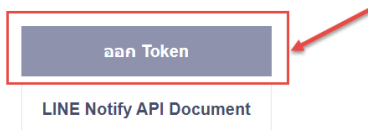
### 3. คลิกที่ชื่อเรา แล้วเลือก หน้าของฉัน



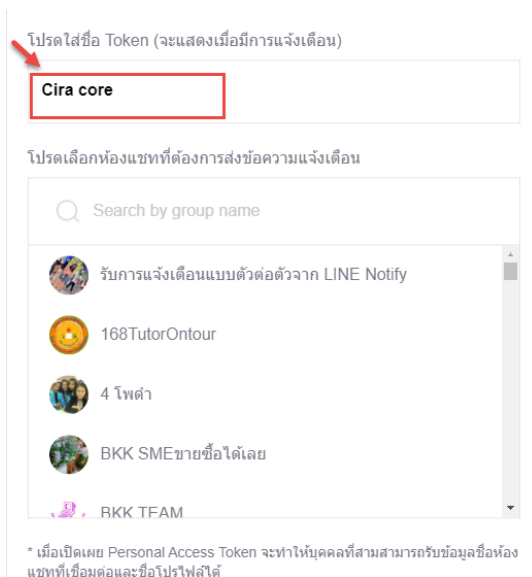
#### 4. ถ้าต้องการอยากจะทำออก Token ใหม่ให้ไปคลิกที่ ออก Token

### ออก Access Token (สำหรับผู้พัฒนา)

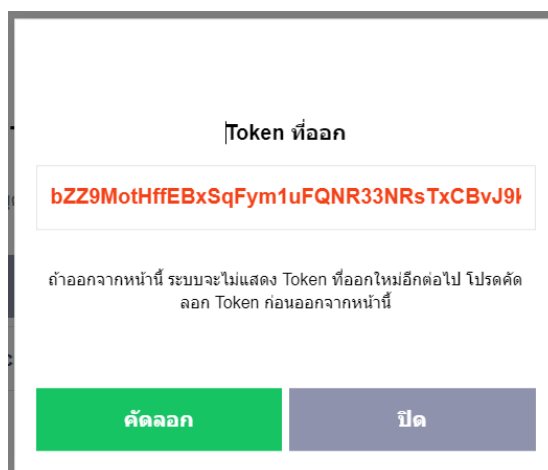
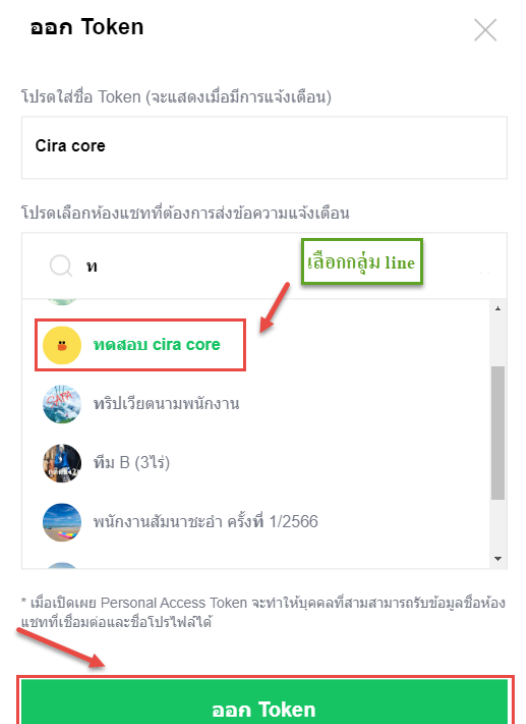
เมื่อใช้ Access Token แบบบุคคล จะสามารถตั้งค่าการแจ้งเตือนได้โดยไม่ต้องลงทะเบียนกับเว็บเซอวิส



#### 5. ทำการตั้งชื่อ ในตัวอย่างจะตั้งชื่อว่า Cira core



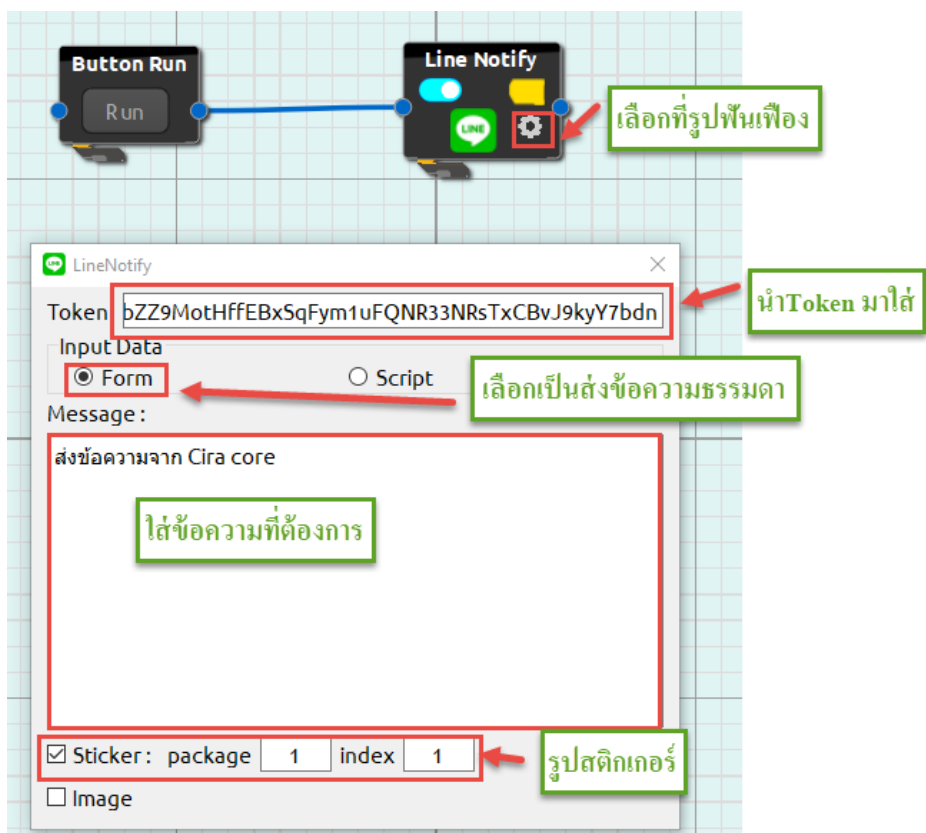
เลือกห้องแชทแนะนำให้สร้างกลุ่มเอาไว้ ด้วยแล้วก็ทำการเพิ่มเพื่อน ให้ line Notify เข้าไปอยู่ในกลุ่มนั้นด้วย แล้วคลิก ที่ปุ่ม ออก Token



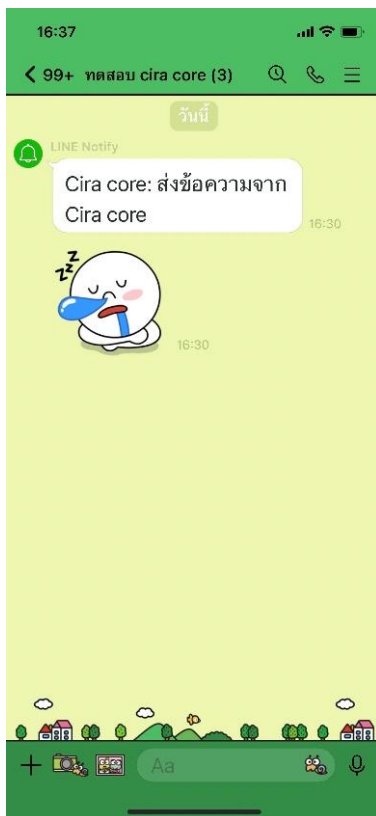
เมื่อเราได้ Token แล้วให้ทำการคัดลอกเก็บไว้ ใน notepad หรือ word เพราะถ้าเราปิดหน้าต่างไปแล้วจะไม่สามารถเปิดขึ้นมาดูได้อีก

## ★ ขั้นตอนการเขียนโปรแกรมใน Cira core

1. ลาก Button Run และ Line Notify มาวาง ลากเส้นเชื่อมความสัมพันธ์
2. เลือกที่รูปฟิ่งเพื่อใส่ Token เลือกรูปแบบการส่ง และพิมพ์ข้อความที่ต้องการส่ง

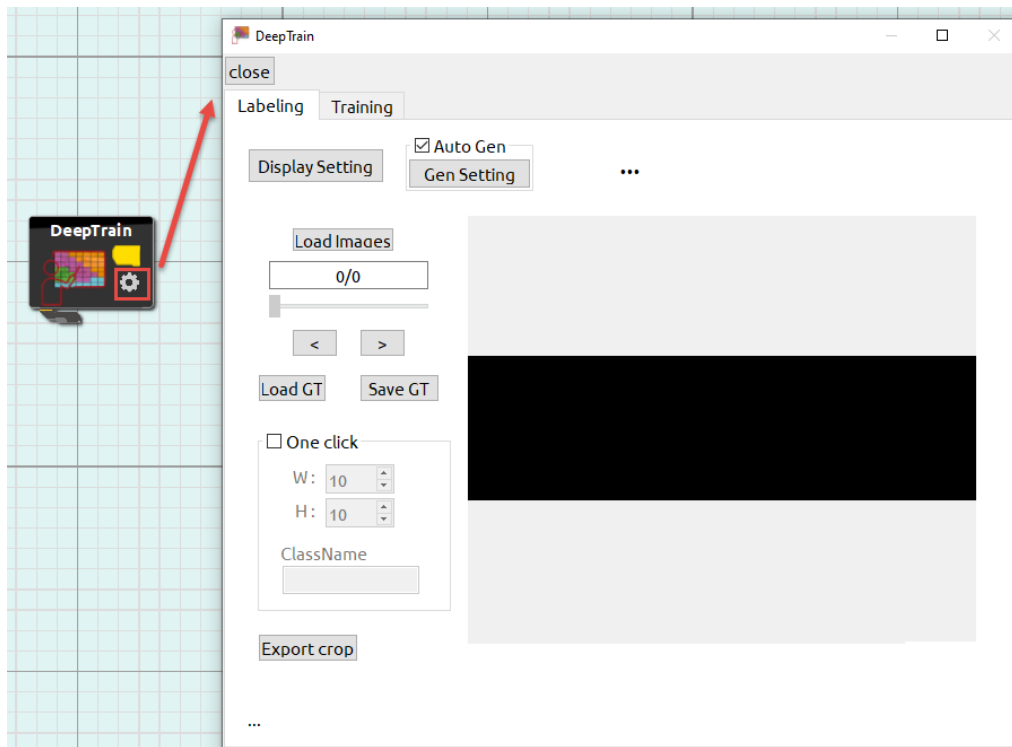


3. กดปิดหน้าต่าง แล้ว กดปุ่ม Run ที่ Button Run โปรแกรมก็จะทำงานส่งข้อความไปยังกลุ่มที่เราเลือกไว้

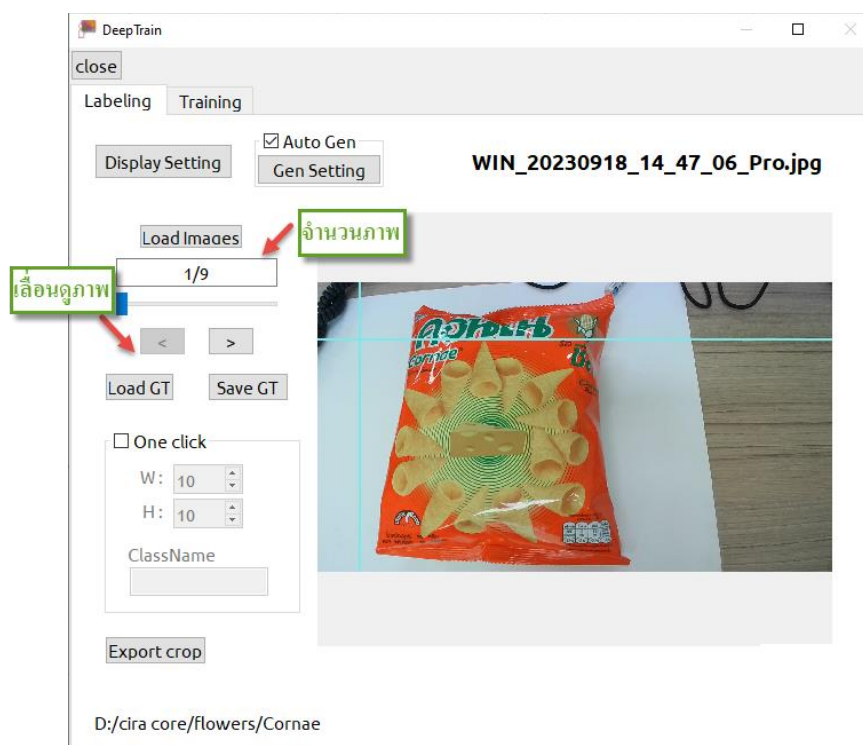


## ★ สร้างโมเดลด้วย DeepTrain

ทำให้คอมพิวเตอร์ของเราได้เรียนรู้และรู้จักกับวัตถุ ด้วยบล็อก DeepTrain ออกมาเพื่อใช้งาน จากนั้นก็สรุปสั้นเพียง



1. จะต้องทำ labeling หรือการติดป้ายกำกับให้กับวัตถุที่เรานำมาสอนให้ Cira core สร้างเป็นโมเดลให้เตรียมรูปภาพที่จะนำมา Train
2. คลิกที่ Load images เลือกรูปที่ต้องการเข้ามา

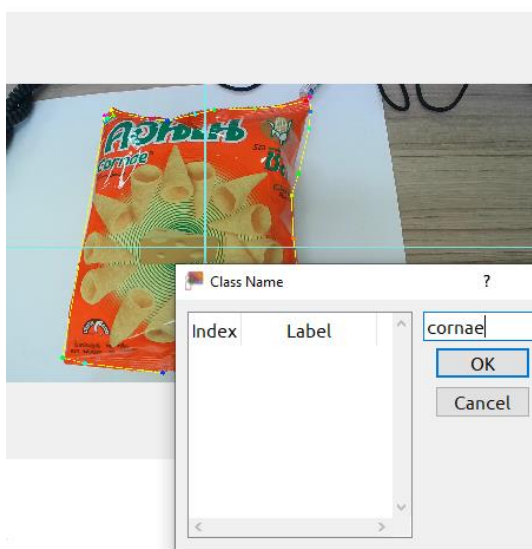




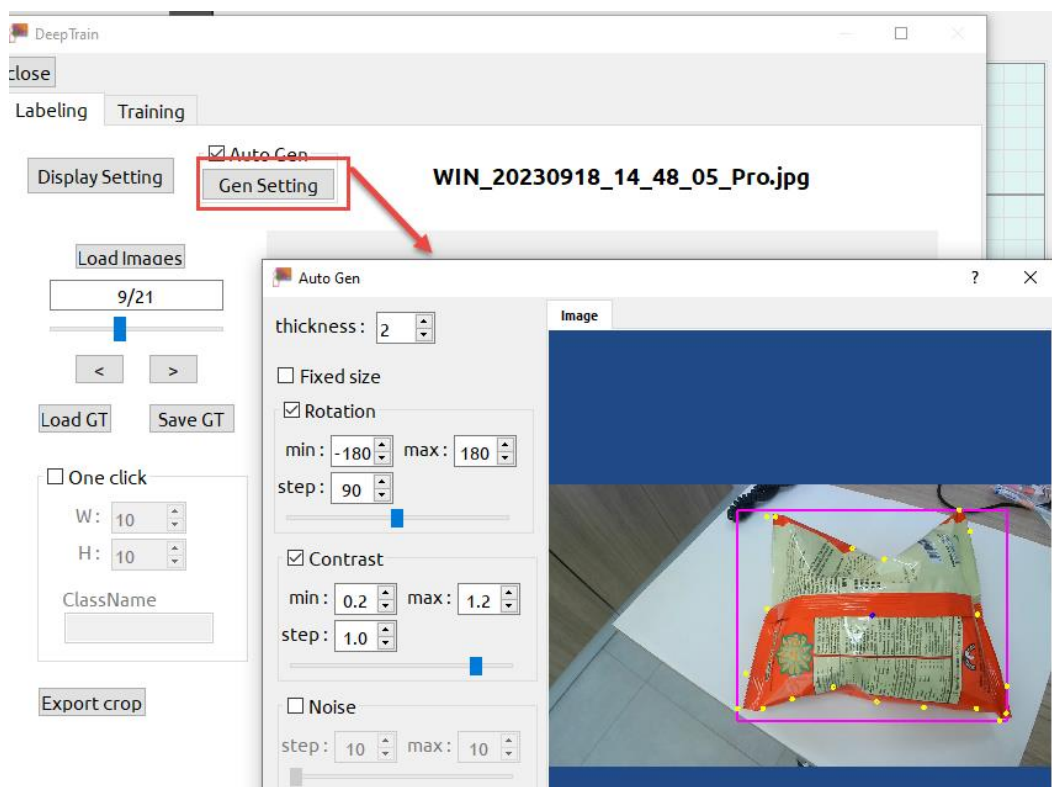
3. ทำการติกรอบให้กับรูปภาพ หรือเลือกพื้นที่ 1. โดยการคลิกเมาส์แล้วลากพื้นที่ไปเรื่อย ๆ 2. ติเป็นกรอบสี่เหลี่ยมเลย 3. วิธีนี้ก็จะเป็นการกำหนดค่าเอง (ใช้ scroll ในการซูมเข้าซูมออกได้)



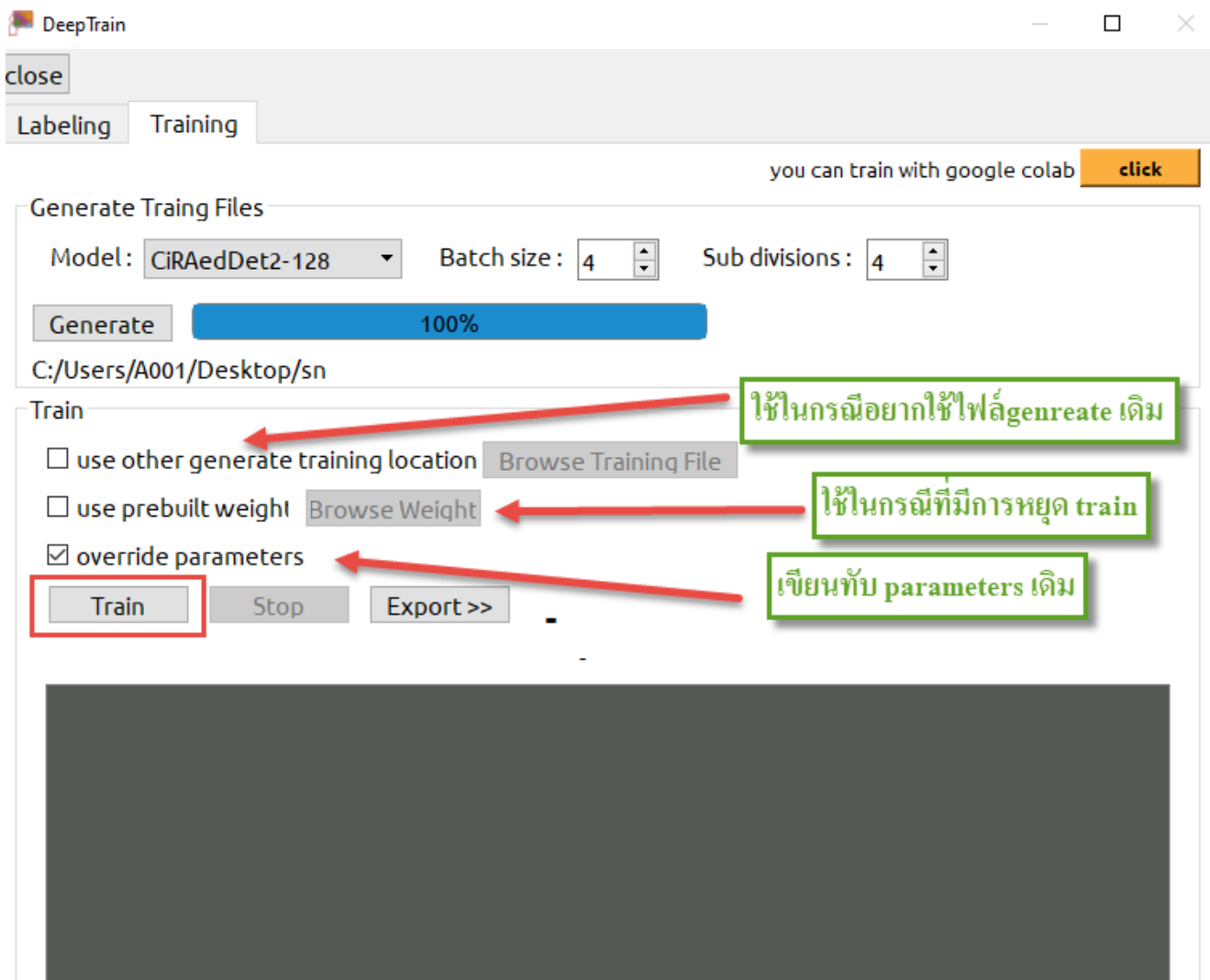
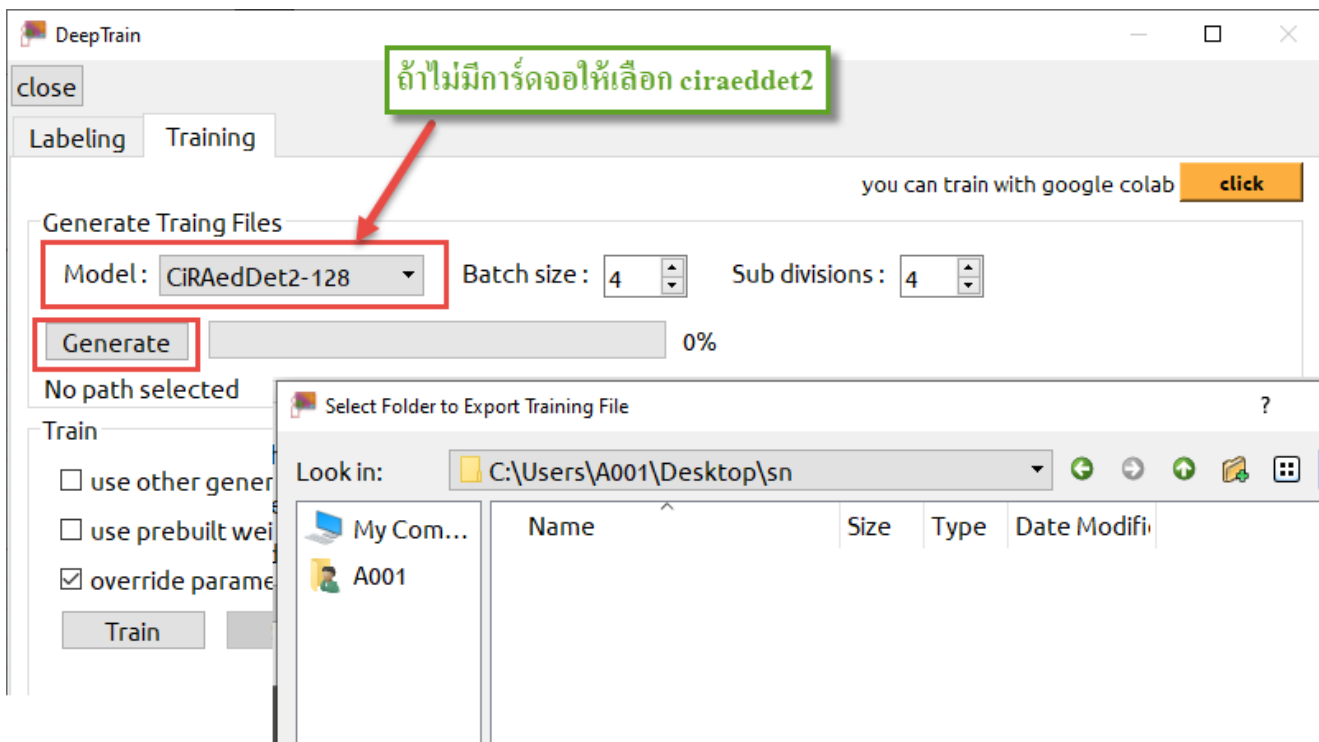
4. เมื่อเสร็จให้กด ' จะ มี หน้าต่าง ให้ เรา กำหนด ป้าย ชื่อ ใส่ ชื่อ แล้วคลิกที่ปุ่ม Ok ทำให้ครบทุกรูป



5. คลิก Gen Setting เป็นการทำให้ภาพเรามีหลายมุมมองมากขึ้น ทำให้ AI ฉลาดมากขึ้น

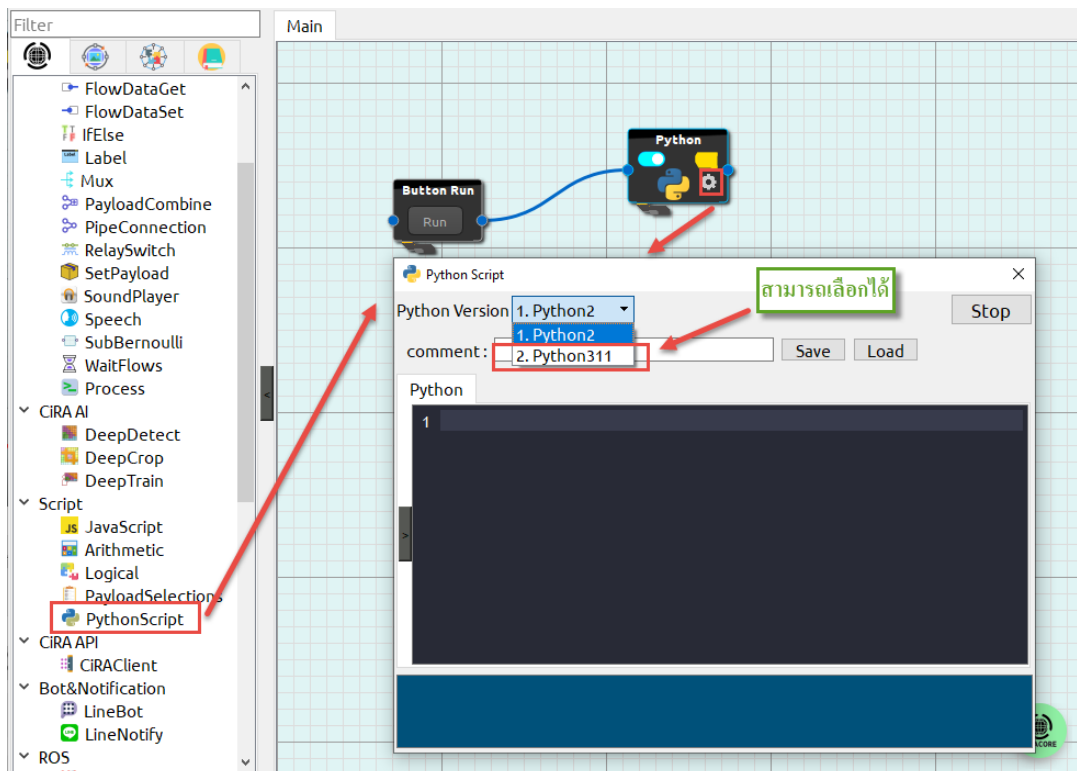


6. เลือก Training ตั้งค่าต่าง ๆ แล้วคลิกที่ Generate เลือกสถานที่เก็บ ต้องเก็บไว้ที่ Drive C เท่านั้น

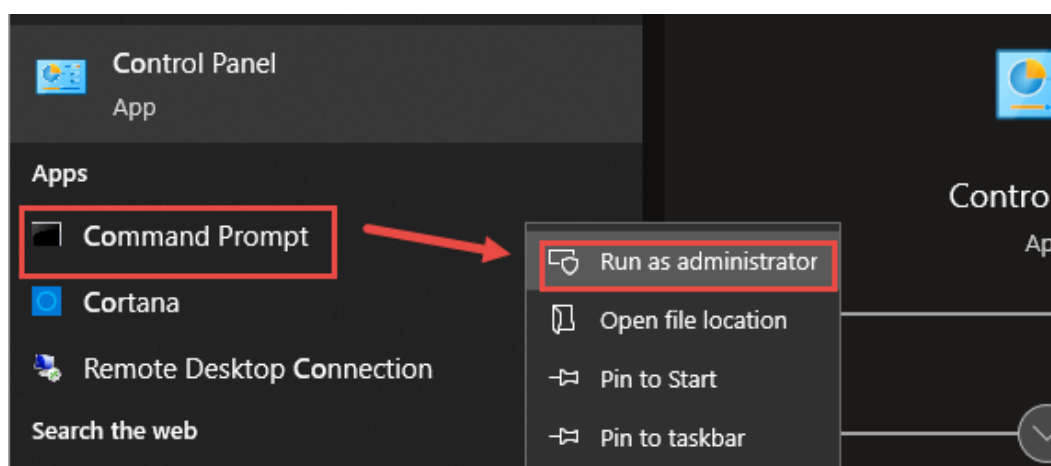


## ★ ตั้งค่าให้ใช้งาน python เวอร์ชัน 3 บน CIRA CORE

1. ในตัว Cira core จะใช้ python 2 แต่ถ้าเราอยากใช้งาน python ที่อยู่ด้านนอก จะต้องมีการตั้งค่า (ตอนลง Python จะต้องทำการคลิกเลือก Add Python 3.11 to Path ด้วยก่อนการ Install)
2. ใช้บล็อก Button Run , Python กดรูปฟันเฟืองเพื่อตรง Python เพื่อเลือกเป็น Python311



4. จะต้องทำการติดตั้ง numpy และ opencv บน python เพื่อให้ cira core รู้จัก ให้เปิด command prompt คลิกขวา เลือก Run as administrator



พิมพ์ cd ตามด้วย path ของ Scripts แล้วพิมพ์คำสั่ง pip install numpy

```

Microsoft Windows [Version 10.0.19041.610]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>cd C:\Users\A001\AppData\Local\Programs\Python\Python310\Scripts
C:\Users\A001\AppData\Local\Programs\Python\Python310\Scripts>pip install numpy
Collecting numpy
  Downloading numpy-1.26.0-cp310-cp310-win_amd64.whl (15.8 MB)
----- 15.8/15.8 MB 5.7 MB/s eta 0:00:00
Installing collected packages: numpy
Successfully installed numpy-1.26.0

[notice] A new release of pip available: 22.3.1 -> 23.2.1
[notice] To update, run: python.exe -m pip install --upgrade pip
C:\Users\A001\AppData\Local\Programs\Python\Python310\Scripts>

```

ติดตั้ง opencv-python พิมพ์คำสั่ง pip install opencv-python

```

C:\Users\A001\AppData\Local\Programs\Python\Python311\Scripts>pip install opencv_python
Requirement already satisfied: opencv_python in c:\users\a001\appdata\roaming\python\python311\site-packages (4.8.0.76)
Requirement already satisfied: numpy>=1.21.2 in c:\users\a001\appdata\roaming\python\python311\site-packages (from opencv_python) (1.25.2)

C:\Users\A001\AppData\Local\Programs\Python\Python311\Scripts>

```

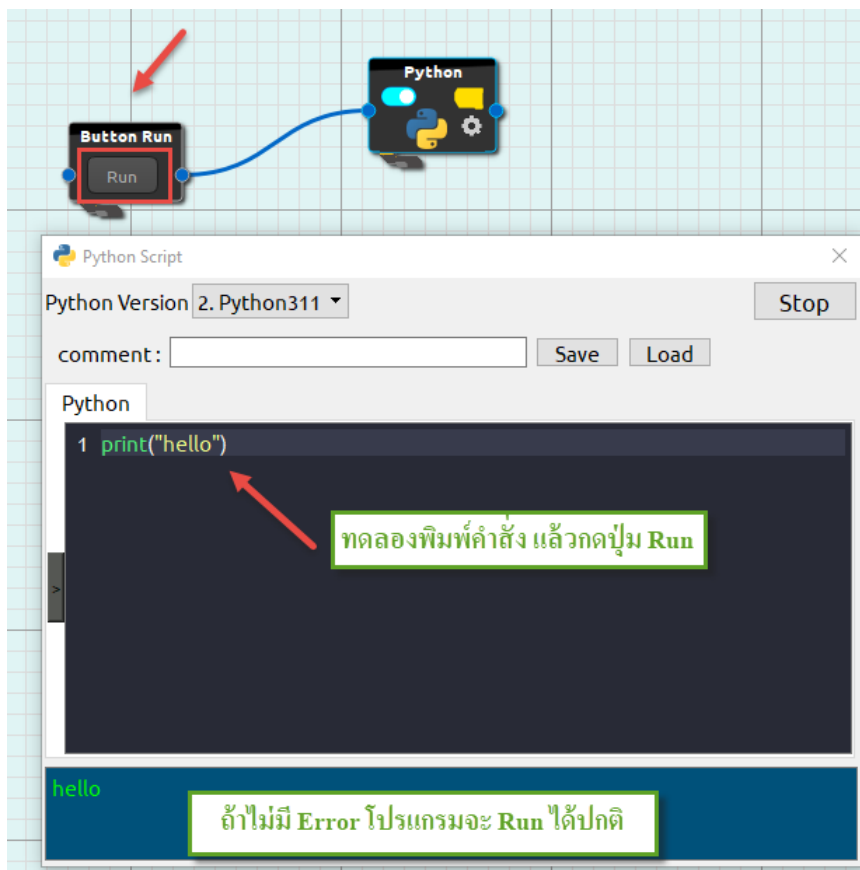
ตัวที่ 3 ติดตั้ง opencv-contrib-python พิมพ์คำสั่ง pip install opencv-contrib-python

```

C:\Users\HP22\AppData\Local\Programs\Python\Python311\Scripts>pip install opencv-contrib-python
Collecting opencv-contrib-python
  Obtaining dependency information for opencv-contrib-python from https://files.pythonhosted.org/packages/81/3c/bbb3ceee9fbefc505f98c24dafda68c7b3c4f83b6951c0712b4623fe4cce/opencv_contrib_python-4.8.1.78-cp37-abi3-win_amd64.whl.metadata
  Downloading opencv_contrib_python-4.8.1.78-cp37-abi3-win_amd64.whl.metadata (20 kB)
Requirement already satisfied: numpy>=1.21.2 in c:\users\hp22\appdata\local\programs\python\python311\site-packages (from opencv-contrib-python) (1.26.0)
  Downloading opencv_contrib_python-4.8.1.78-cp37-abi3-win_amd64.whl (44.8 MB)
----- 44.8/44.8 MB 24.2 MB/s eta 0:00:00
Installing collected packages: opencv-contrib-python
Successfully installed opencv-contrib-python-4.8.1.78

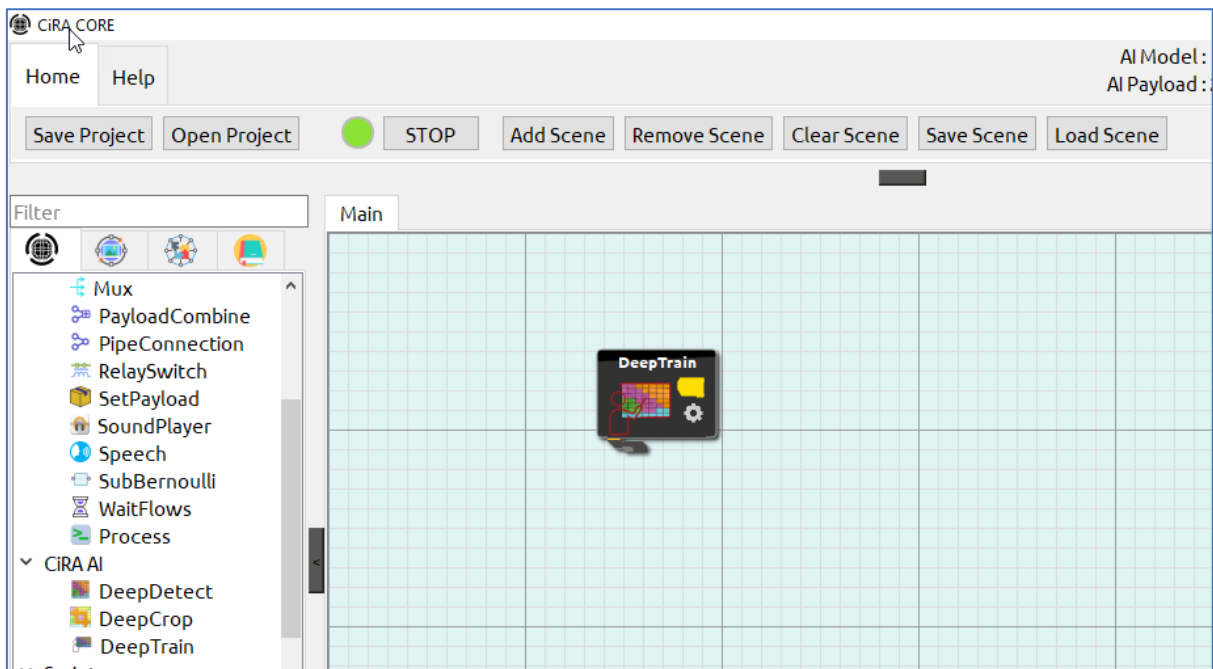
```

ทดสอบ

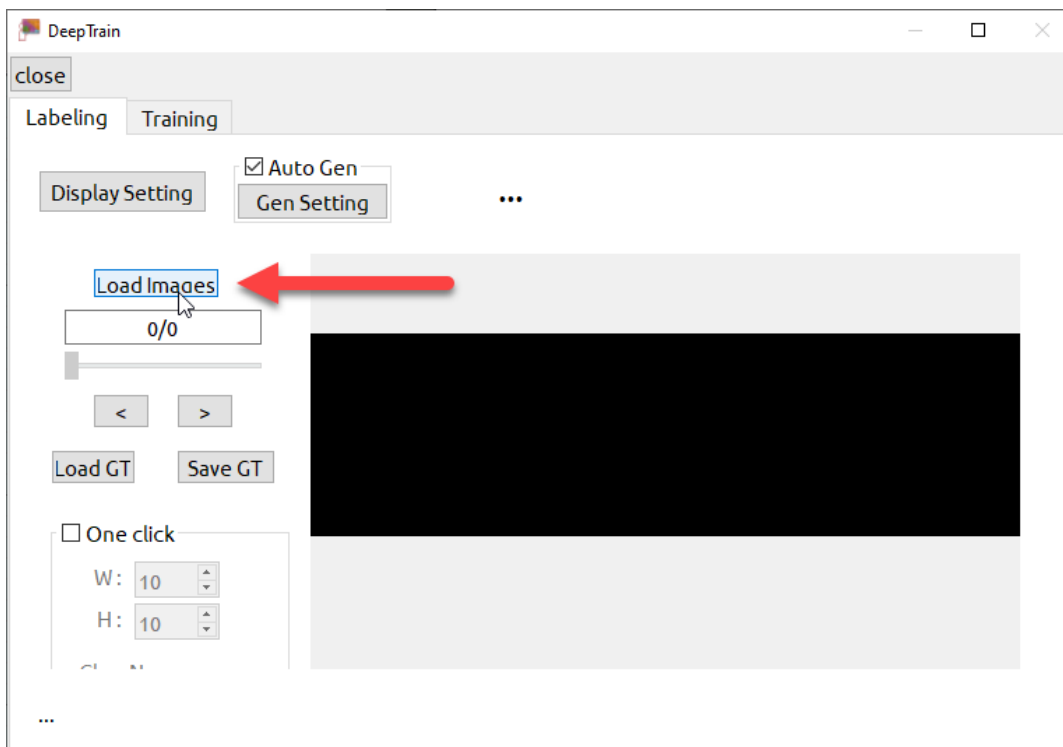


## ขั้นตอนการ DeepTrain จาก Google Colab

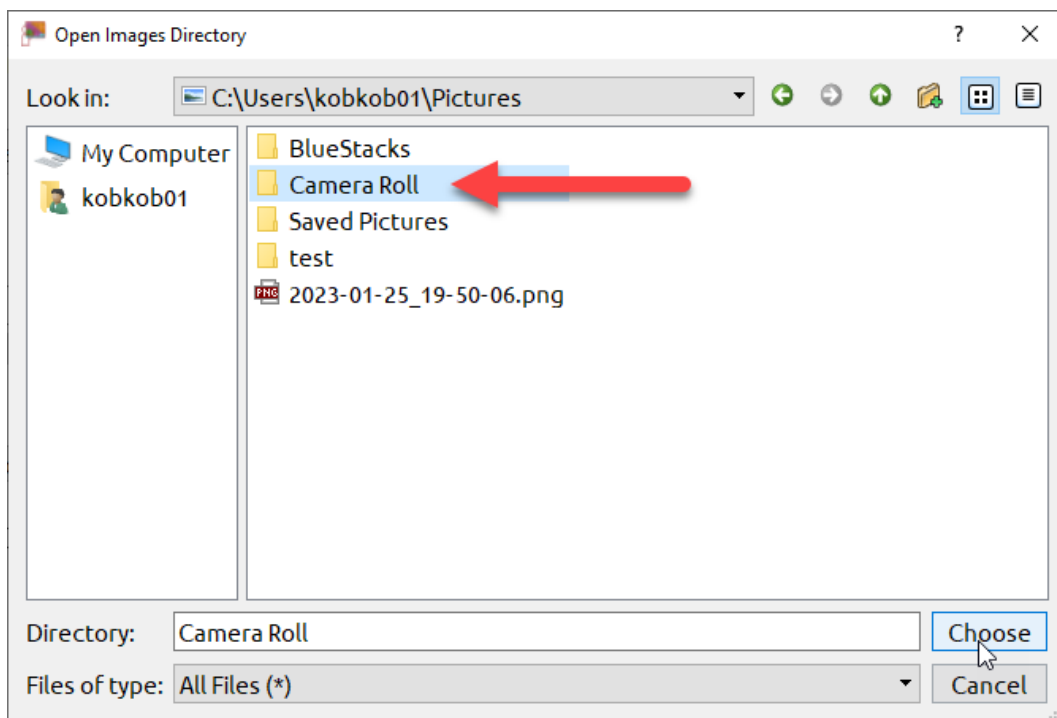
### 1. ลากบล็อก DeepTrain



2. คลิกรูปพื้นเฟืองที่บล็อก DeepTrain จากนั้นติดย้ายกำกับรูปภาพ โดยคลิกปุ่ม Load Images

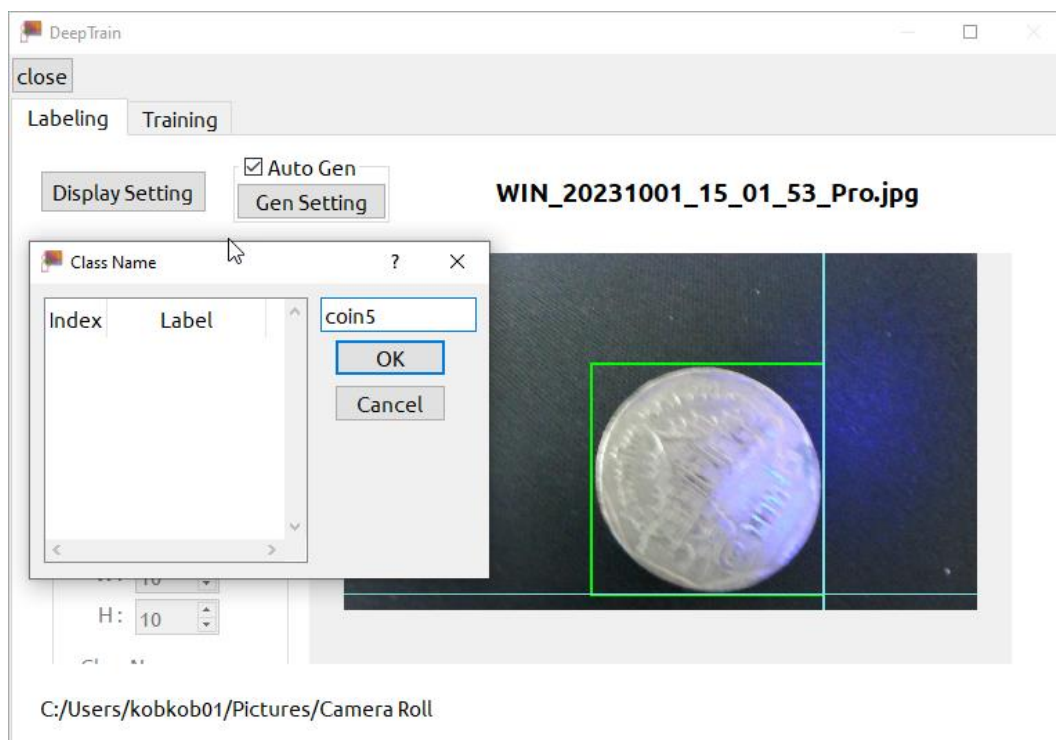


3. เข้าไปในโฟลเดอร์ Pictures/Camera Roll จากนั้นคลิกปุ่ม Choose เพื่อเลือกรูปภาพที่ถ่ายรูปวัตถุ

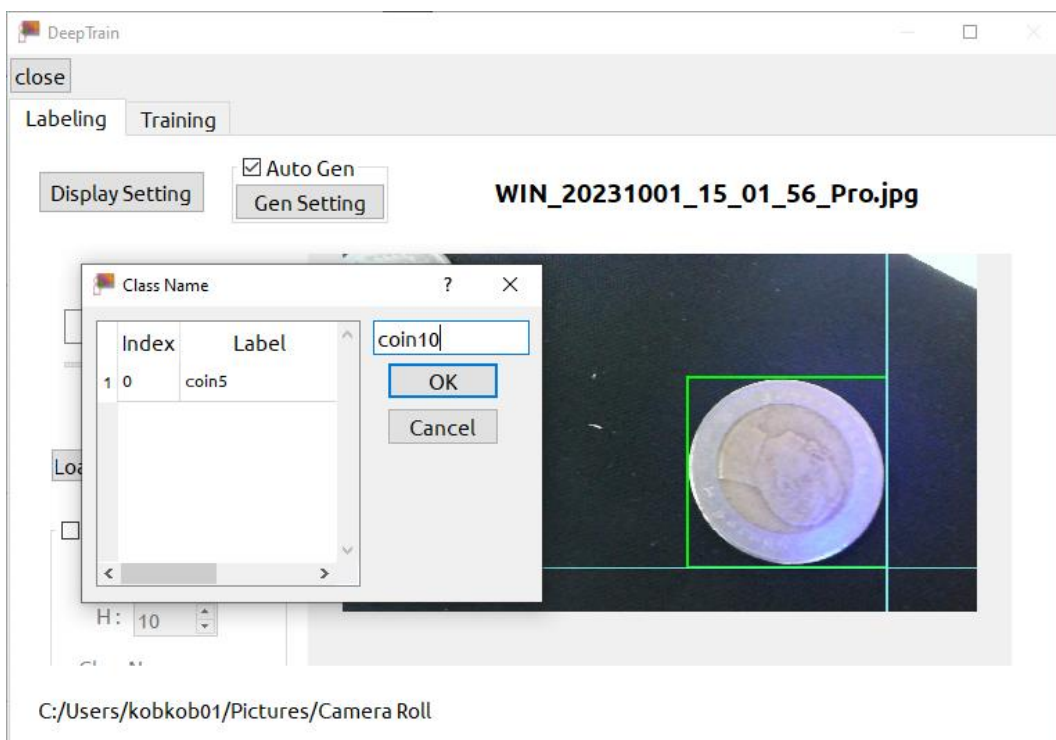


4. ลากกรอบสี่เหลี่ยมล้อมรอบเหรียญ 5 บาท จากนั้นใส่ชื่อลาเบลเป็น “coin5” แล้วคลิก OK

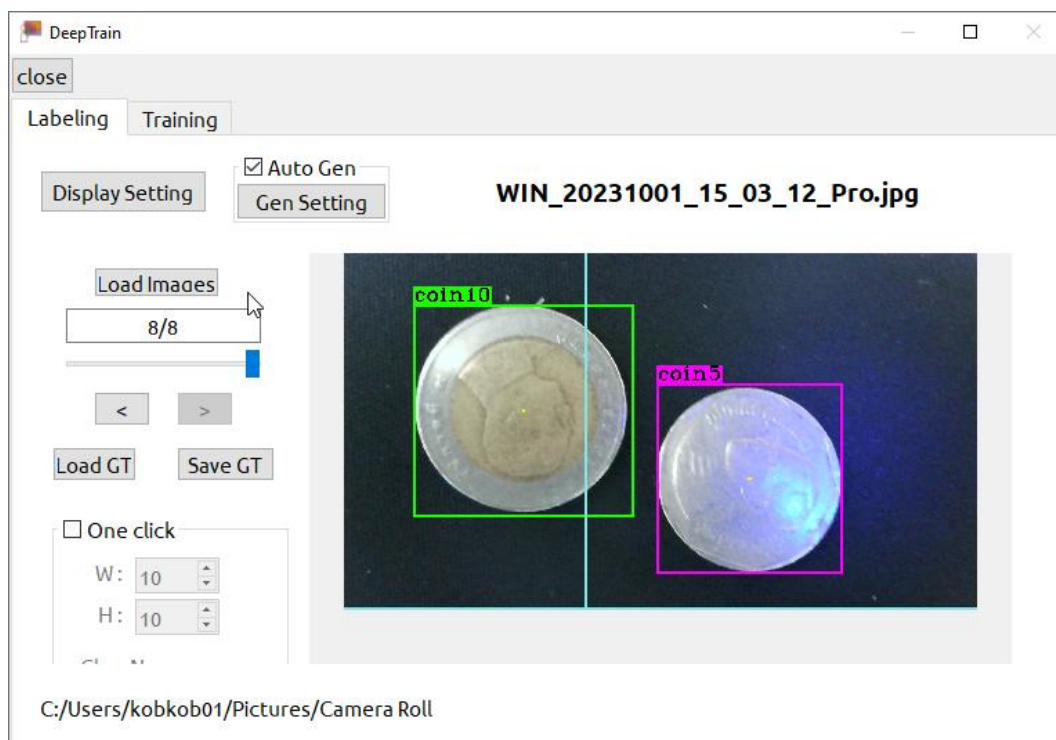




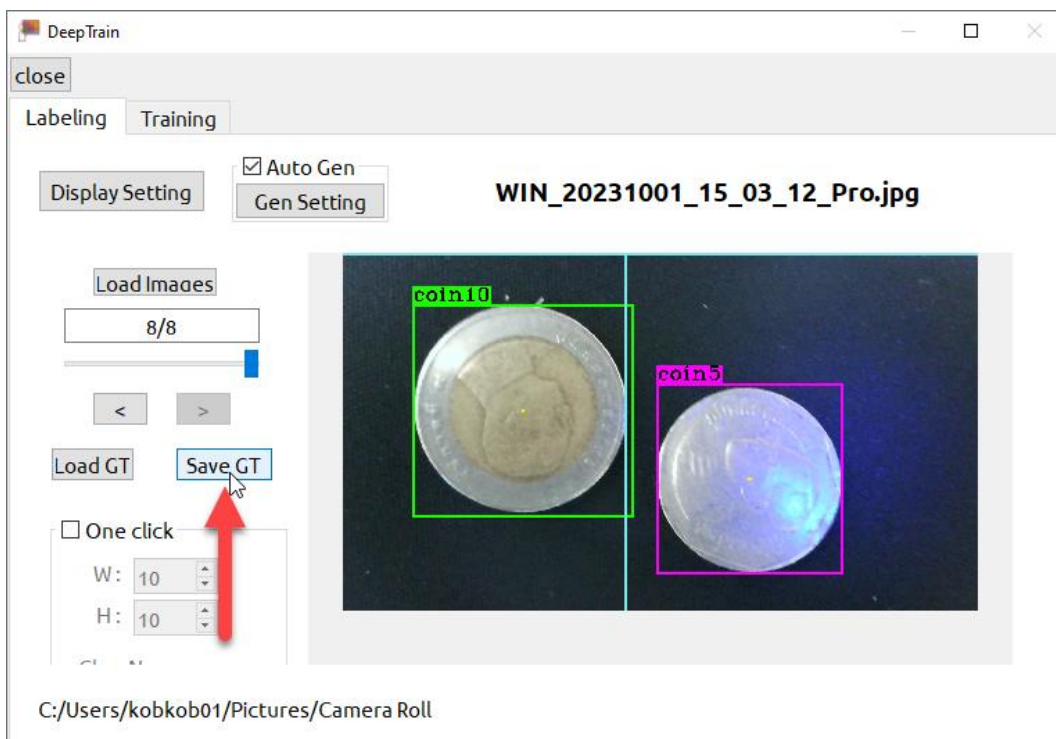
5. ลากกรอบสี่เหลี่ยมล้อมรอบเหรียญ 10 บาท จากนั้นใส่ชื่อลาเบลเป็น “coin10” แล้วคลิก OK



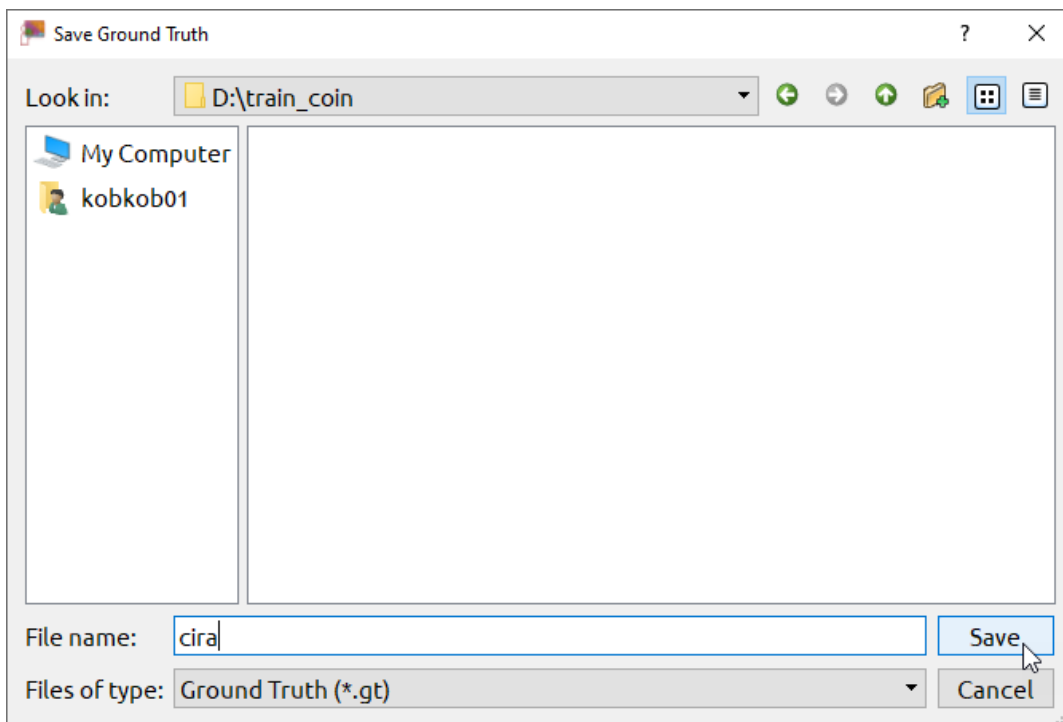
6. หากในรูปมีเหรียญมากกว่า 1 เหรียญ ให้ลากกรอบสี่เหลี่ยมล้อมรอบเหรียญนั้น แล้วใส่ชื่อลาเบลเป็น ให้ถูกต้อง แล้วคลิก OK



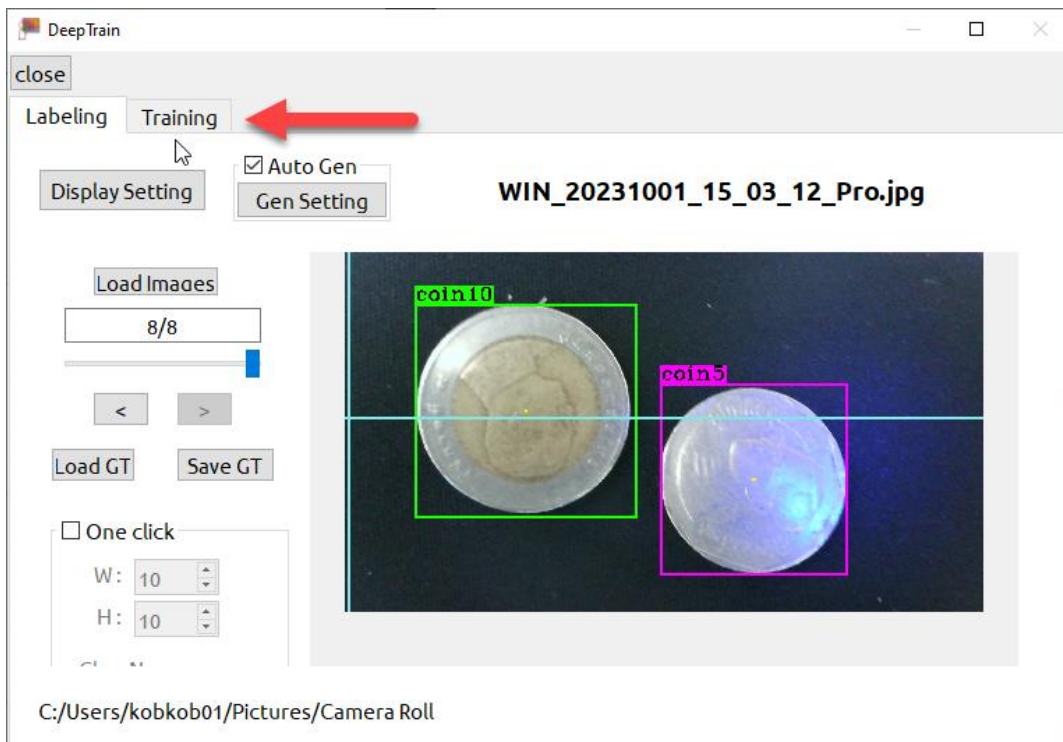
7. เมื่อติดย้ายกำกับวัตถุครบทุกรูปแล้ว ให้คลิกปุ่ม Save GT เพื่อบันทึกไฟล์ .gt ที่ใช้สำหรับการเทรนวัตถุ



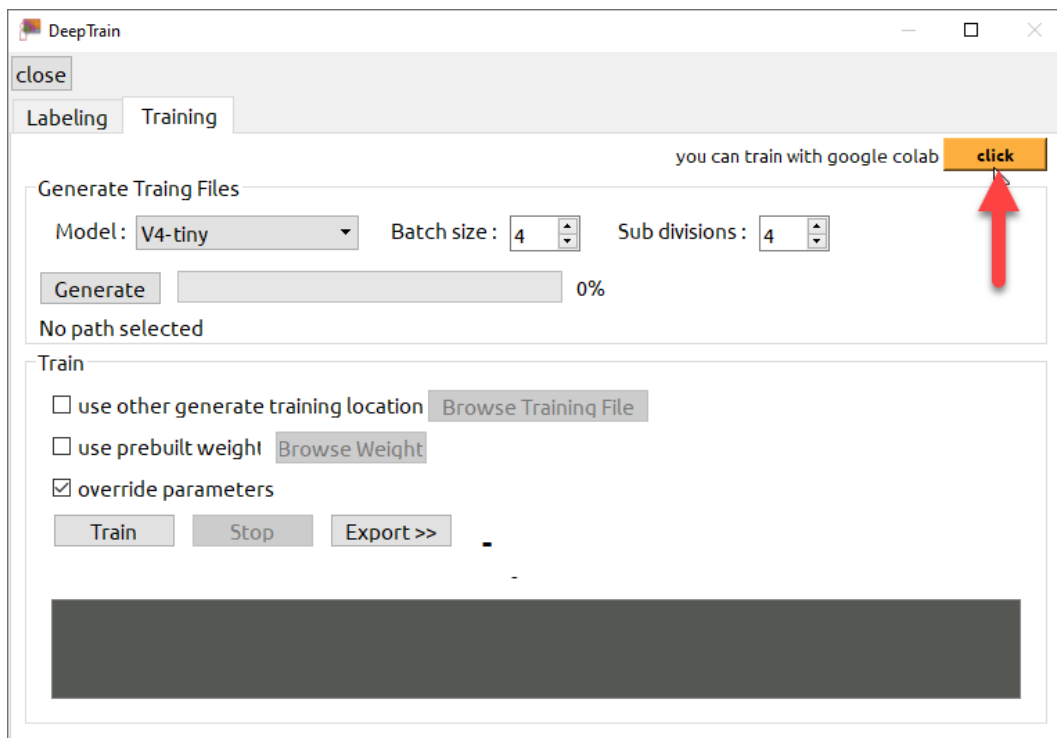
8. บันทึกไฟล์ .gt ไว้ในไดรฟ์ D:/ โดยตั้งชื่อไฟล์ในช่อง File name จากนั้นคลิกปุ่ม Save



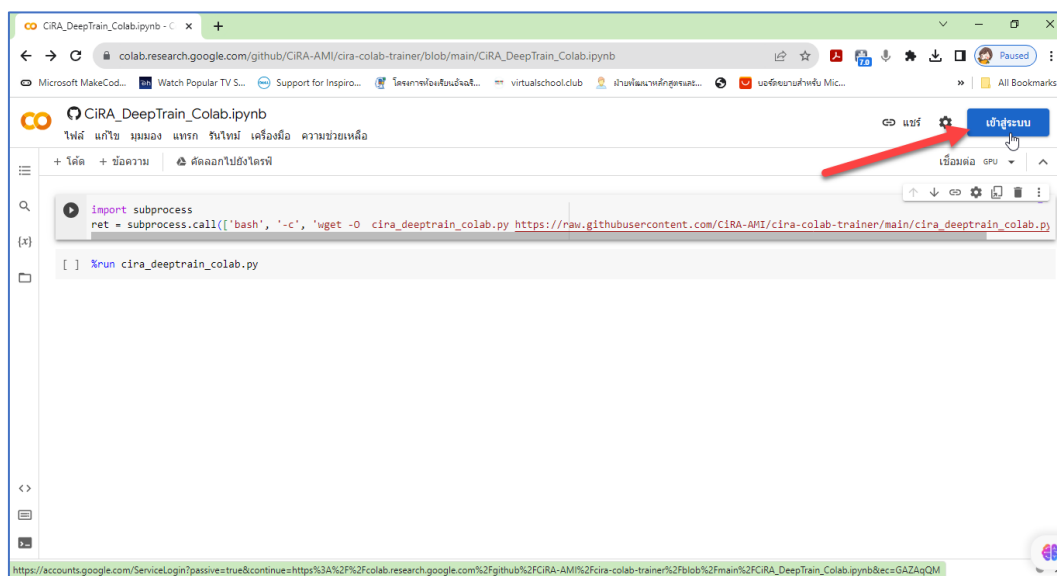
9. เริ่มต้นการเทรนวัตถุ โดยคลิกแท็บ Training



## 10. เลือกเทรนผ่าน Google Colab โดยคลิกที่ปุ่ม click



## 11. ล็อกอินเข้าสู่ระบบ โดยคลิกปุ่ม “เข้าสู่ระบบ”



12. กรอกอีเมล G-Mail ลงไป แล้วคลิกปุ่ม “ถัดไป”

Google

ลงชื่อเข้าใช้

ใช้บัญชี Google ของคุณ

อีเมลหรือโทรศัพท์

168edu.training@gmail.com

หากไม่มีอีเมล

หากไม่ใช่คอมพิวเตอร์ของคุณ ให้ใช้โหมดผู้มาเยือนเพื่อลงชื่อเข้าใช้แบบส่วนตัว ดูข้อมูลเพิ่มเติม

สร้างบัญชี

ถัดไป

13. กรอกรหัสผ่าน แล้วคลิกปุ่ม “ถัดไป”

Google

168education 168education

168edu.training@gmail.com

ป้อนรหัสผ่าน

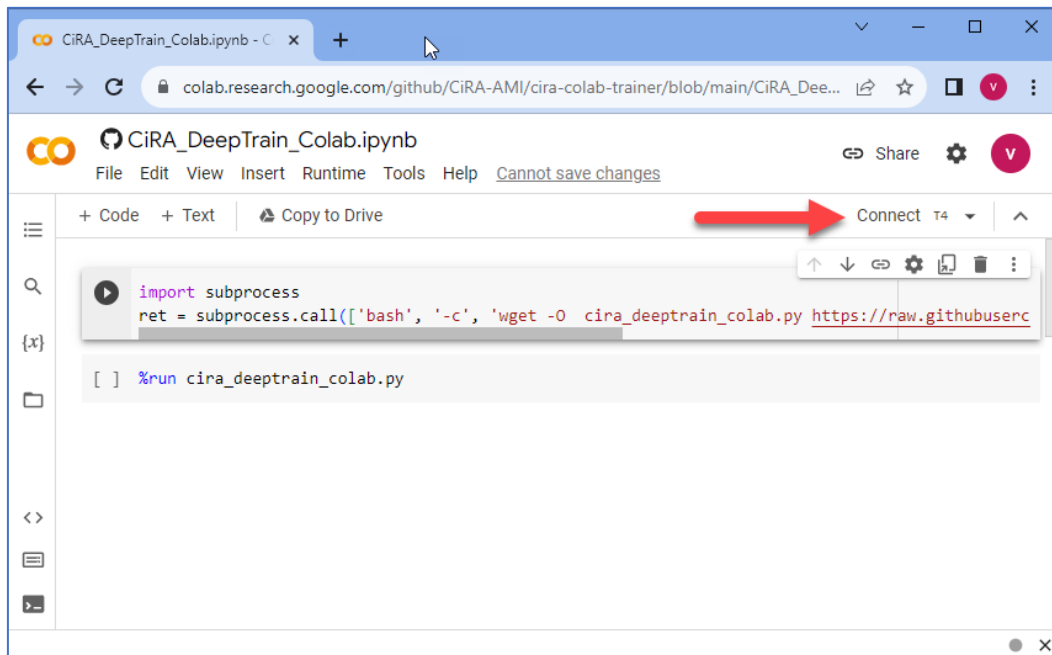
.....

แสดงรหัสผ่าน

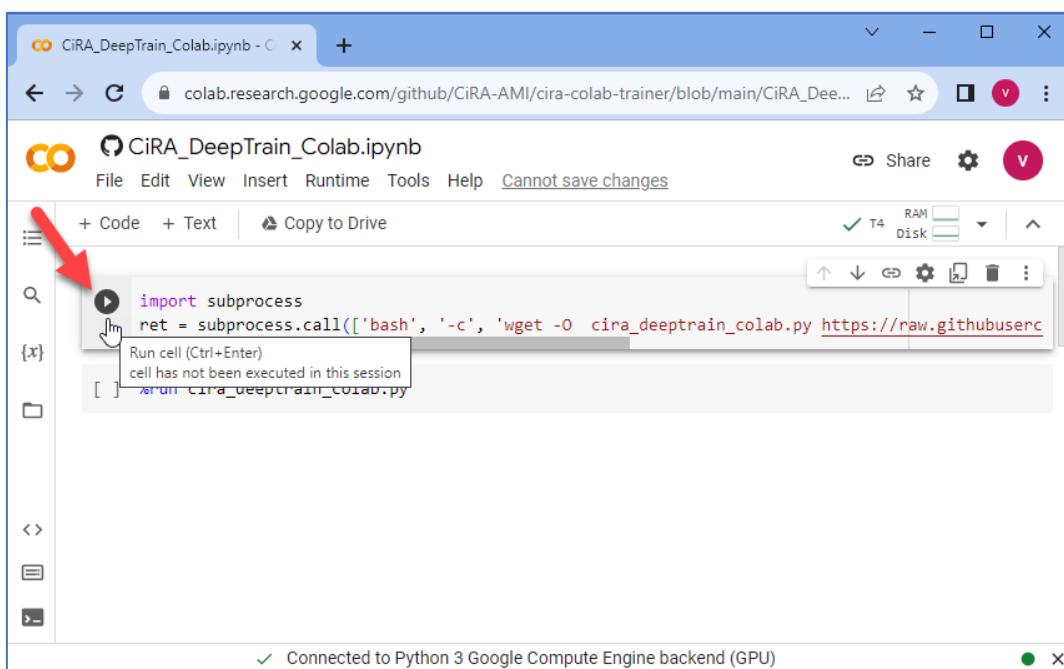
หากไม่มีรหัสผ่าน

ถัดไป

#### 14. คลิกปุ่ม Connect

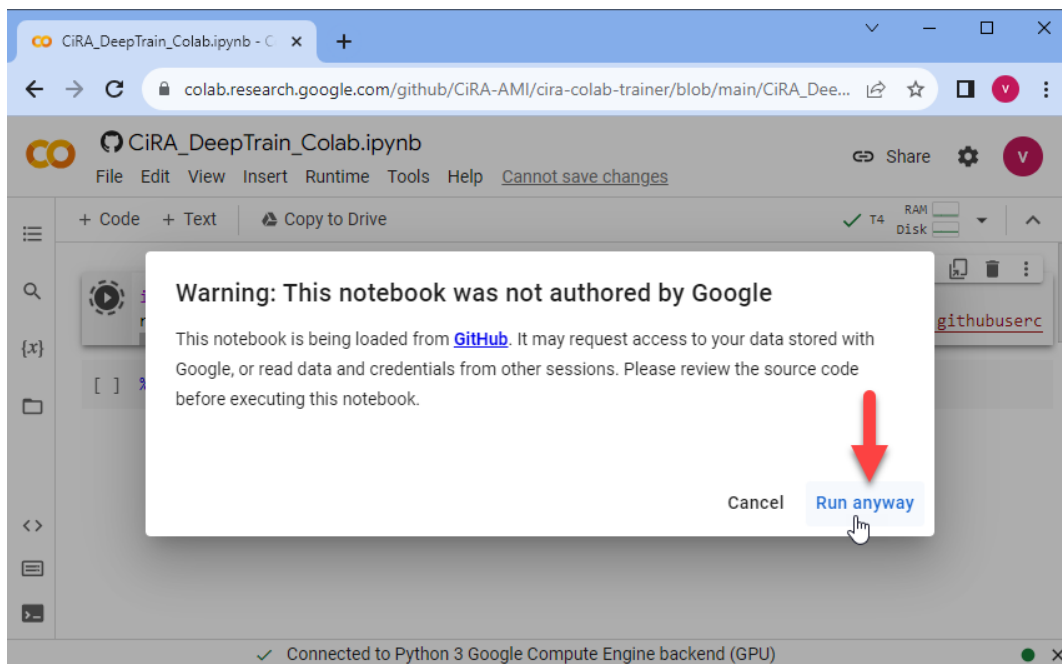


#### 15. คลิกปุ่มรันในโค้ดคำสั่งแรก

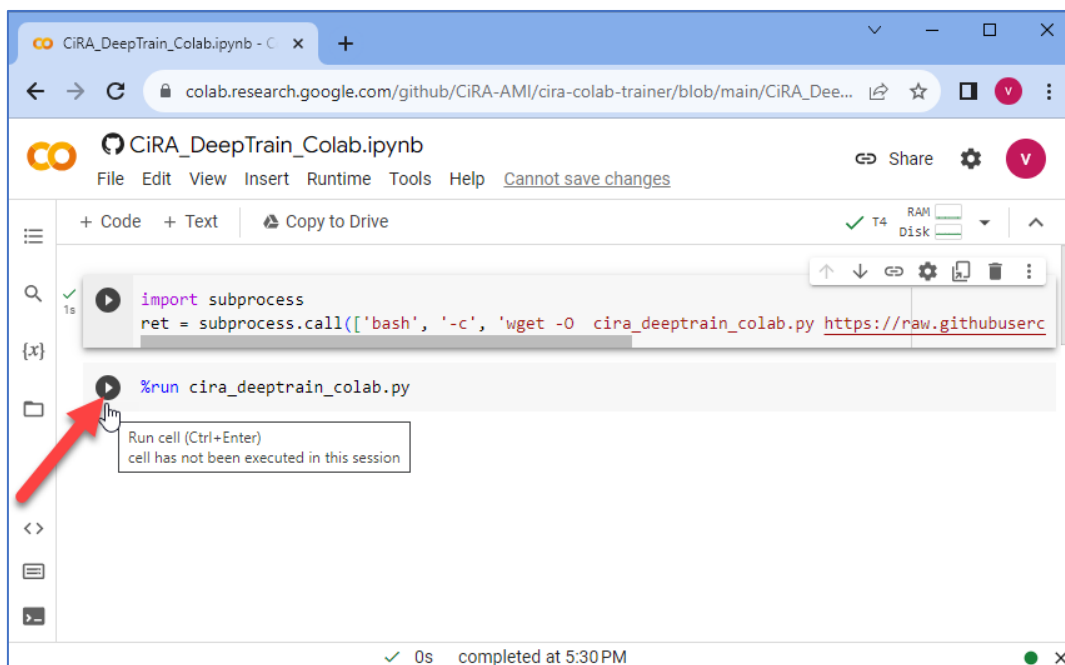




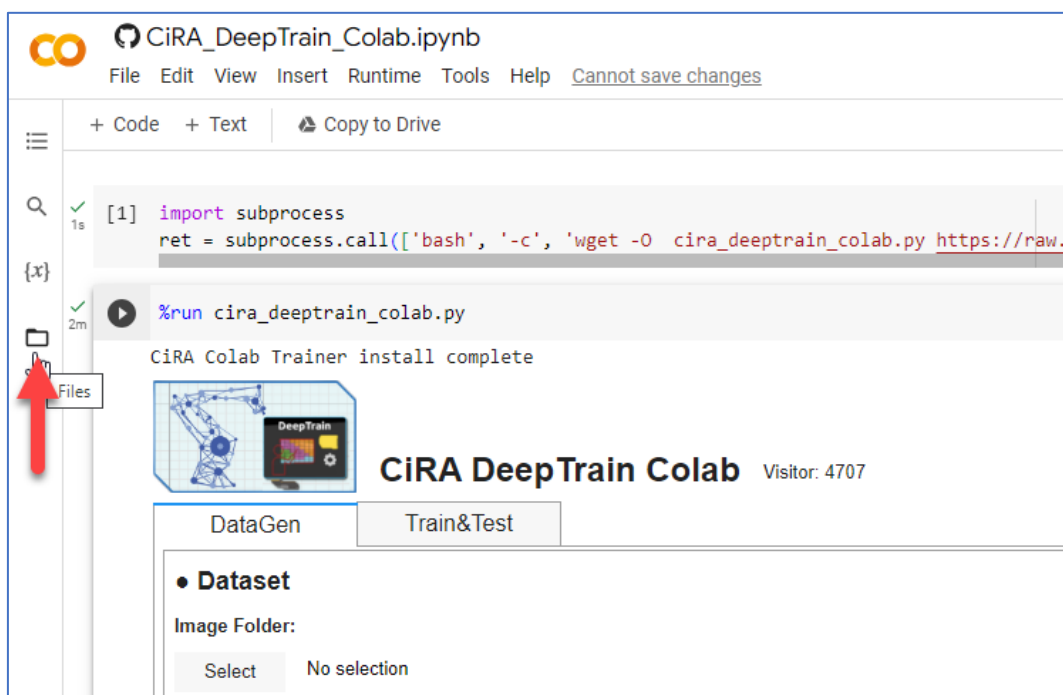
## 16. คลิกปุ่ม Run anyway



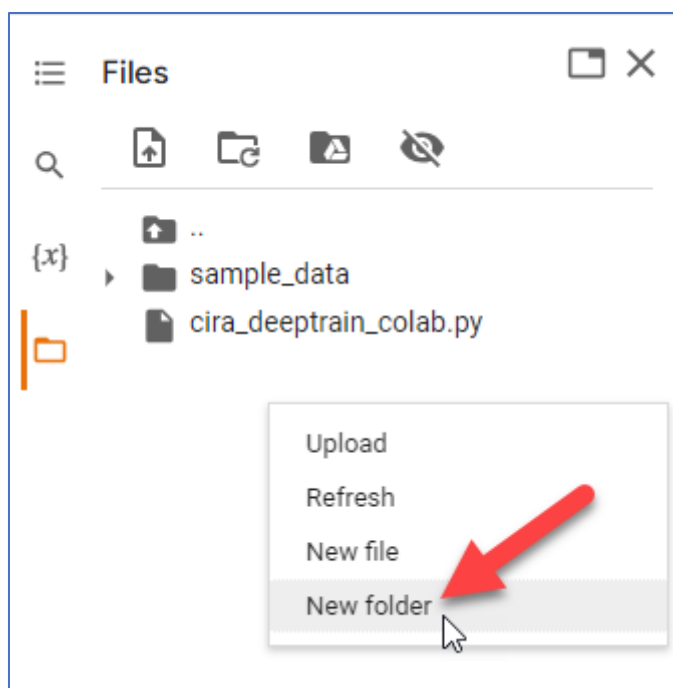
## 17. รันโค้ดคำสั่งที่สอง



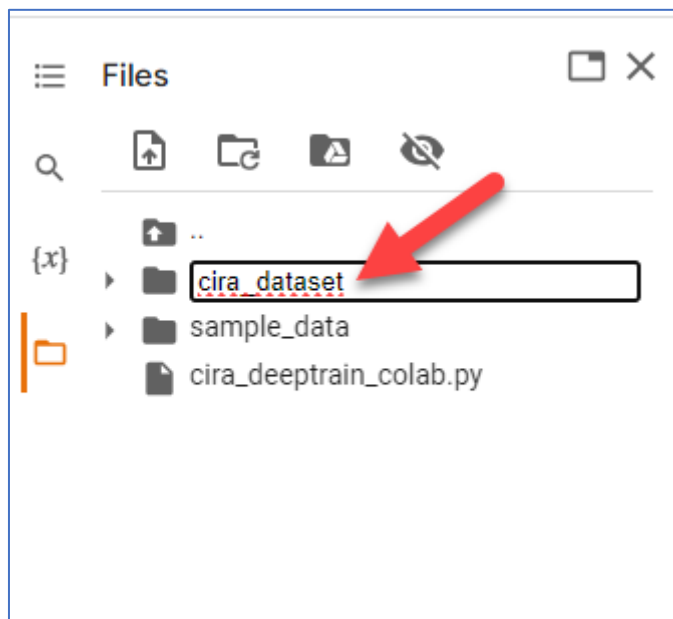
18. คลิกที่ Files เพื่อเปิดดูไฟล์เดอร์ข้อมูล



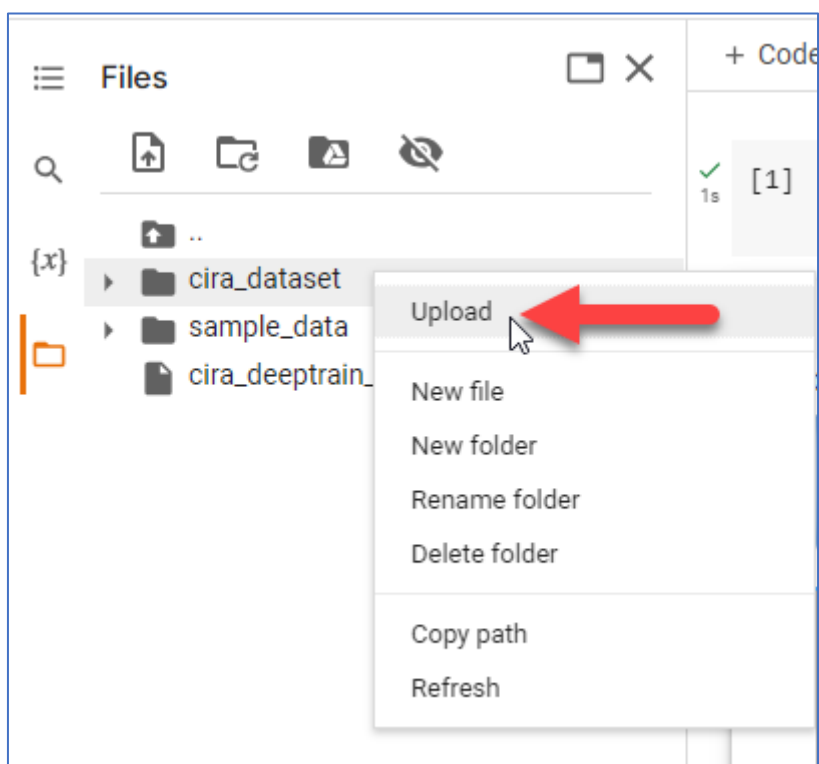
19. สร้างไฟล์เดอร์ใหม่ขึ้นมาเพื่อเก็บไฟล์ Dataset โดยคลิกขวาที่พื้นที่ว่าง จากนั้นเลือก New folder



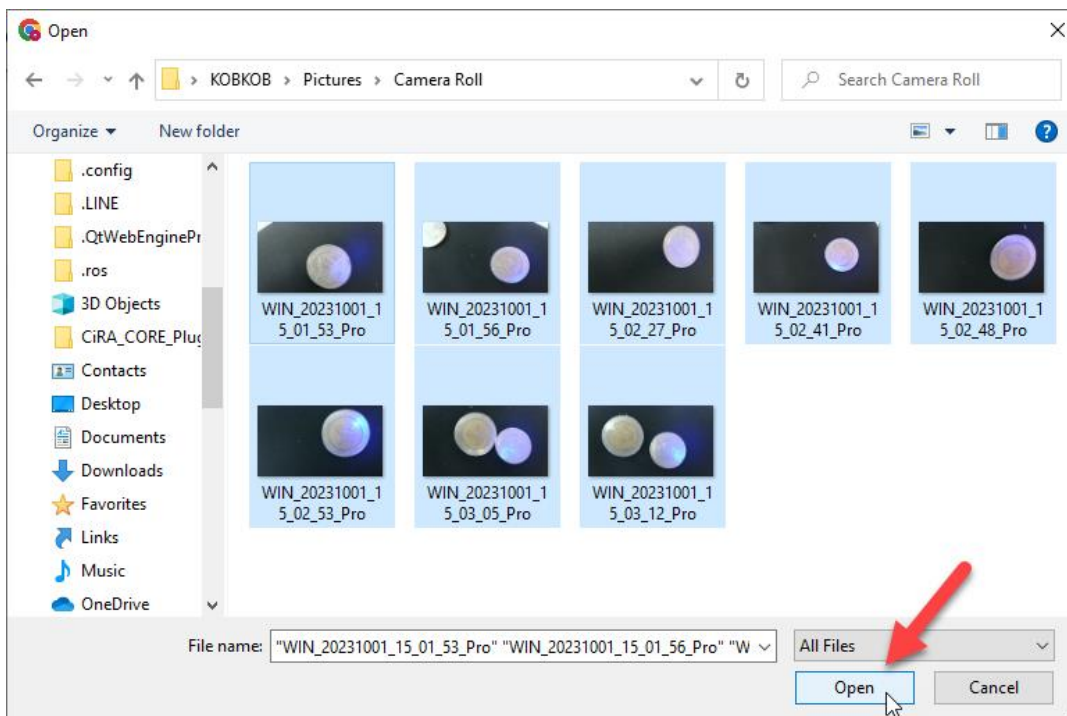
20. ตั้งชื่อโฟลเดอร์ว่า cira\_dataset



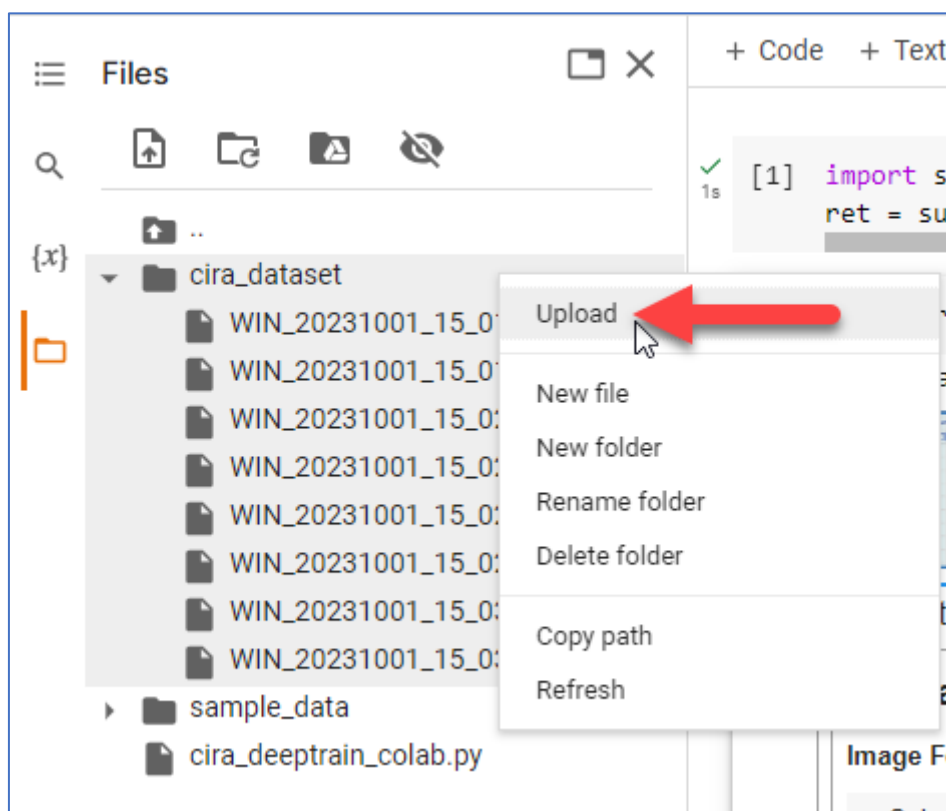
21. คลิกจุด 3 จุดด้านหลังโฟลเดอร์ cira\_dataset จากนั้นเลือก Upload



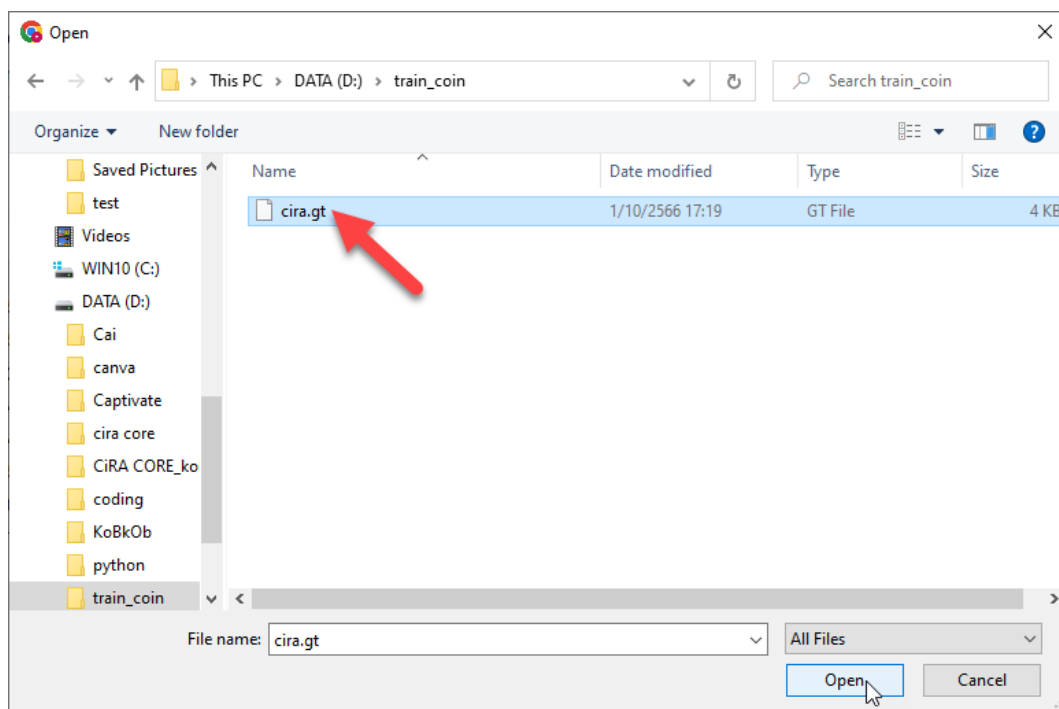
22. อัปโหลดรูปภาพเหรียญ โดยคลิกเลือกไฟล์รูปภาพทั้งหมดในโฟลเดอร์ Camera Roll แล้วคลิกปุ่ม Open



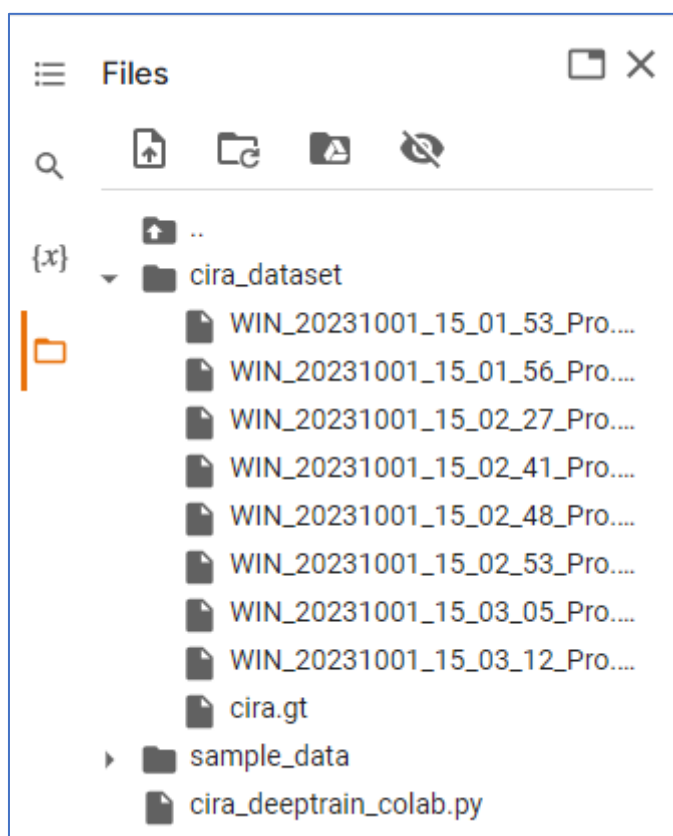
23. อัปโหลดไฟล์ .gt โดยคลิกจุด 3 จุดด้านหลังโฟลเดอร์ cira\_dataset อีกครั้ง จากนั้นเลือก Upload



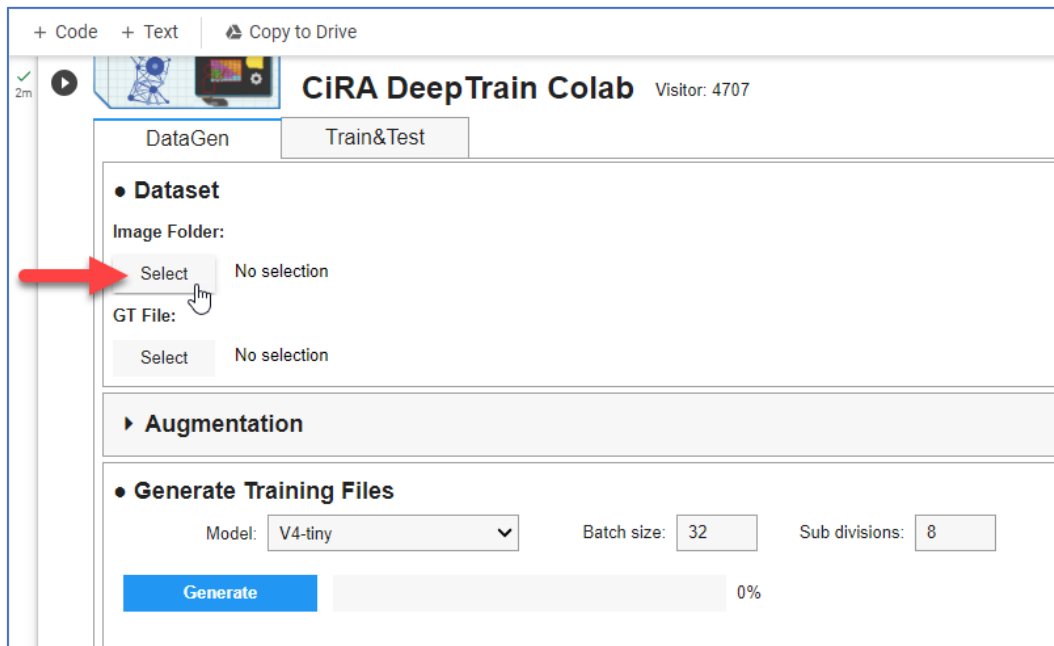
24. คลิกเลือกไฟล์ `cira.gt` ที่อยู่ใน `D:/` แล้วคลิกปุ่ม Open



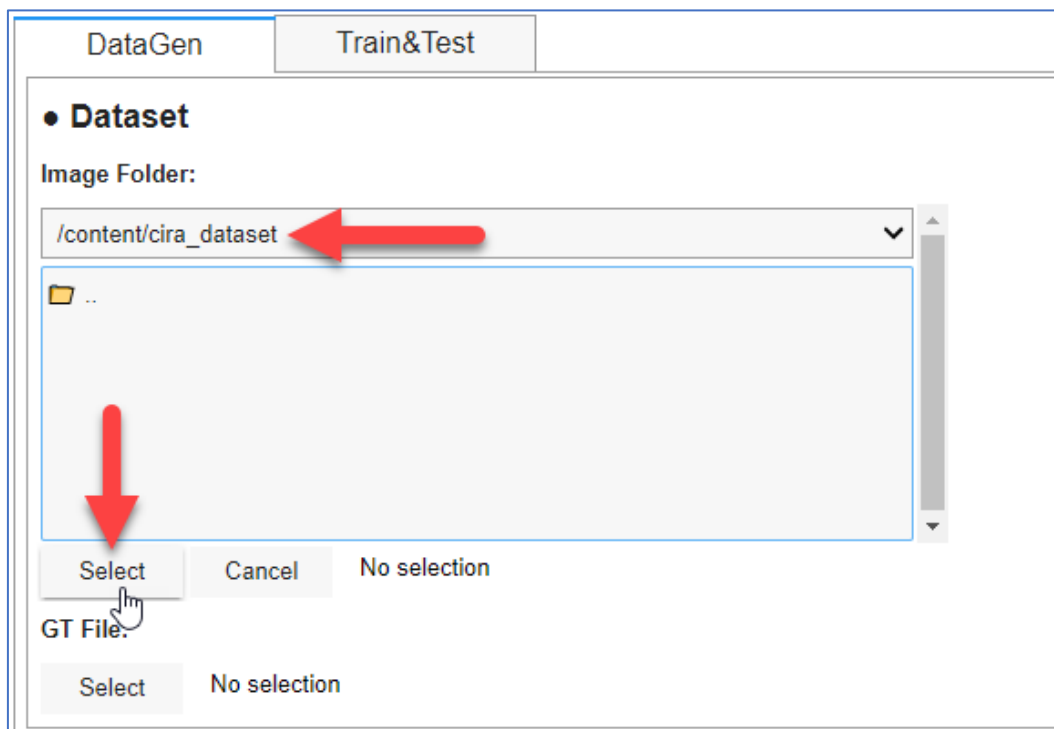
25. จะได้ไฟล์รูปภาพและไฟล์ `.gt` เข้ามาในโฟลเดอร์ `cira_dataset` ดังรูป



26. เลือก Image Folder โดยคลิกปุ่ม Select



27. เลือกไฟล์เดอร์ cira\_dataset จากนั้นคลิกปุ่ม Select





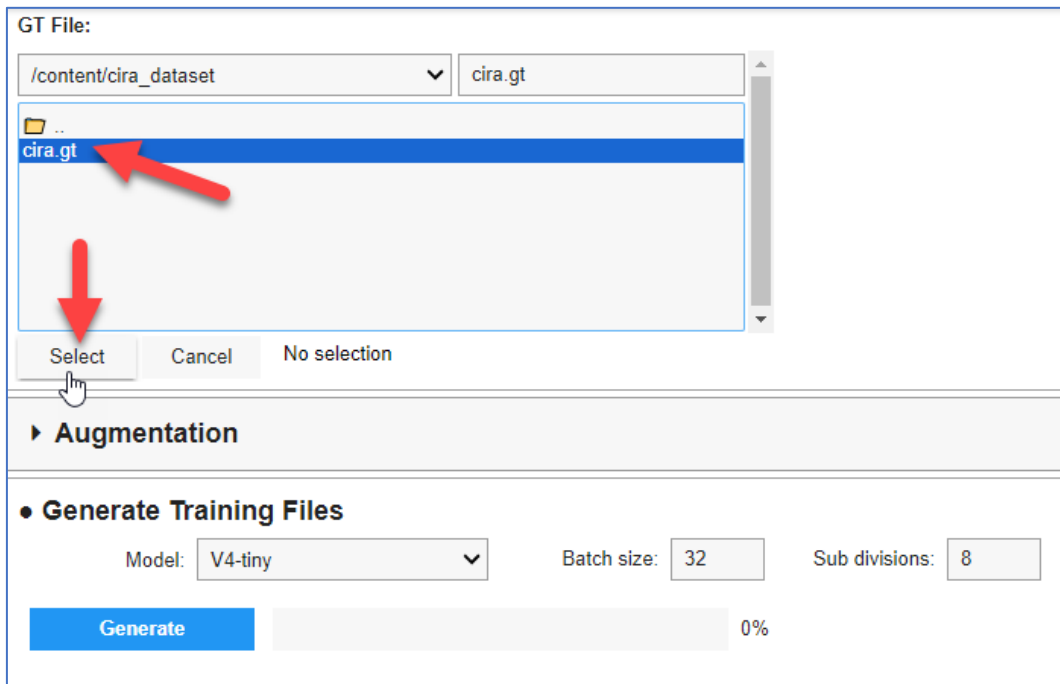
จะได้พาธของ Image Folder ดังรูป

The screenshot shows the 'DataGen' interface with the 'Train&Test' tab selected. Under the 'Dataset' section, the 'Image Folder' is set to '/content/cira\_dataset/' with a 'Change' button to its left. A red arrow points to this path. Below it, the 'GT File' section shows a 'Select' button and the text 'No selection'. The 'Augmentation' section is collapsed. The 'Generate Training Files' section shows 'Model: V4-tiny', 'Batch size: 32', and 'Sub divisions: 8'. A 'Generate' button is present with a progress bar at 0%.

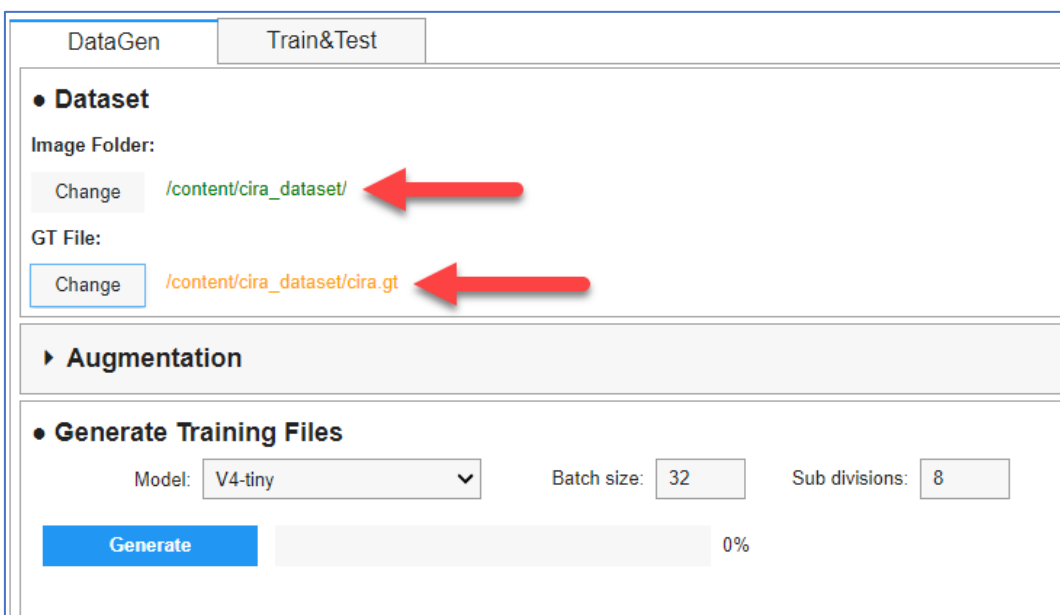
28. เลือก GT File โดยคลิกปุ่ม Select

This screenshot is similar to the previous one, but the 'Select' button in the 'GT File' section is highlighted with a red arrow and a mouse cursor, indicating the next step in the process.

29. เลือกไฟล์ cira\_dataset จากนั้นคลิกปุ่ม Select



จะได้พารามิเตอร์ของ Image Folder และไฟล์ .gt ดังรูป



30. คลิกที่ Augmentation เพื่อเพิ่ม Dataset ให้มากขึ้น โดยการขยายรูปภาพให้หลากหลายกว่าเดิม

The screenshot shows the DataGen interface with two tabs: 'DataGen' and 'Train&Test'. Under the 'DataGen' tab, there are two main sections: 'Dataset' and 'Augmentation'. The 'Dataset' section includes 'Image Folder' (set to '/content/cira\_dataset/') and 'GT File' (set to '/content/cira\_dataset/cira.gt'). The 'Augmentation' section is highlighted with a red arrow and a mouse cursor. Below it, the 'Generate Training Files' section shows 'Model' set to 'V4-tiny', 'Batch size' set to '32', and 'Sub divisions' set to '8'. A 'Generate' button is visible with a progress bar at 0%.

สามารถปรับเปลี่ยนค่าต่างๆ เพื่อให้ได้รูปภาพใหม่ขึ้นมา หรือไม่แก้ไขค่าต่างๆ ก็ได้

The screenshot shows the DataGen interface with the 'Augmentation' section expanded. It contains four sub-sections, each with a checked checkbox and input fields for min, max, and step values:

- Rotation:** min: -180, max: 180, step: 90
- Contrast:** min: 0.2, max: 1.2, step: 1
- Noise:** max: 10, step: 10
- Blur:** max: 9, step: 9

31. คลิกปุ่ม Generate

**Dataset**

Image Folder:

GT File:

**Augmentation**

Rotation  
min:   
max:   
step:

Contrast  
min:   
max:   
step:

Noise  
max:   
step:

Blur  
max:   
step:

**Generate Training Files**

Model:  Batch size:  Sub divisions:

0%

32. คลิกแท็บ Train&Test เพื่อเริ่มต้นเทรน โมเดล

DataGen **Train&Test**

**Dataset**

Image Folder:

GT File:

**Augmentation**

Rotation  
min:   
max:   
step:

Contrast  
min:   
max:   
step:

Noise  
max:   
step:

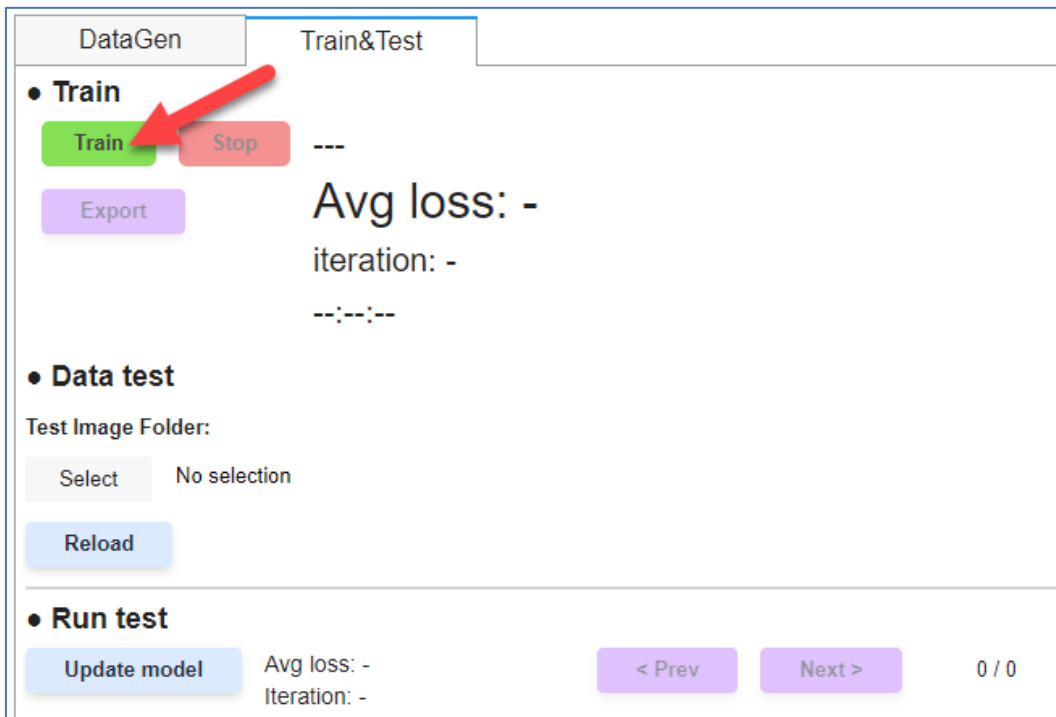
Blur  
max:   
step:

**Generate Training Files**

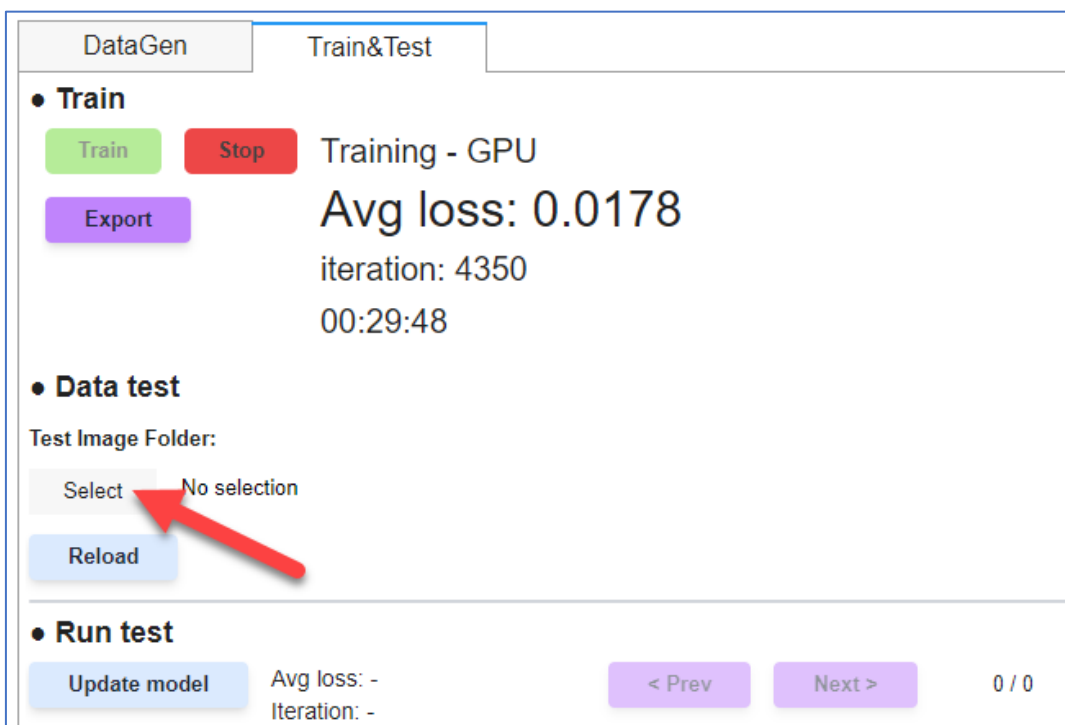
Model:  Batch size:  Sub divisions:

100%

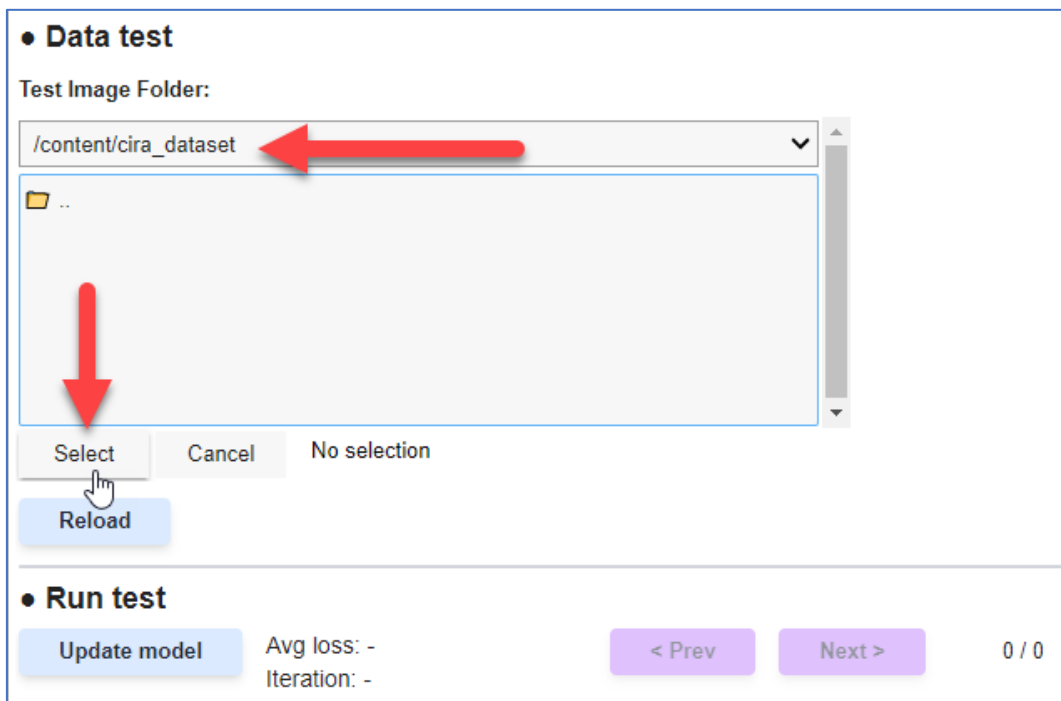
33. คลิกปุ่ม Train สังเกตค่า Avg loss จะเริ่มตั้งแต่ 999 แล้วจะค่อยๆ ลดลงไปเรื่อยๆ



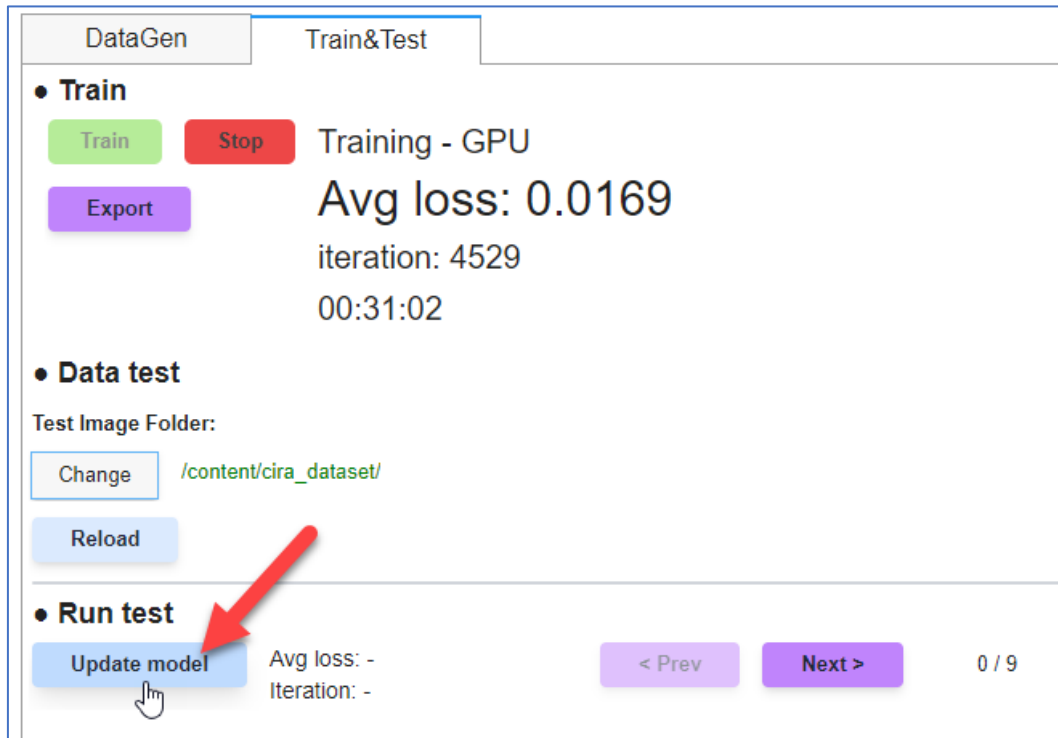
34. เมื่อรอบในการเทรน (iteration) อยู่ที่ประมาณ 2000 รอบ (iteration: 2000) ให้ทดสอบการ Detect รูปภาพ โดยคลิกที่ Select ในส่วนของ Test Image Folder



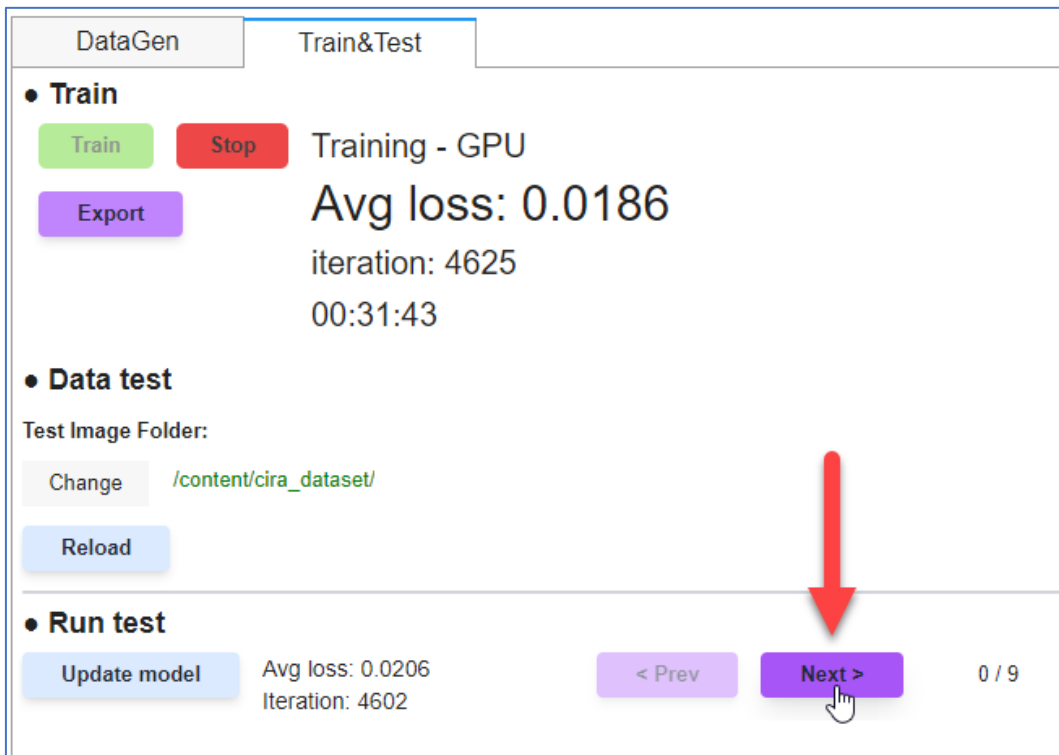
35. เลือกโฟลเดอร์ cira\_dataset จากนั้นคลิก Select



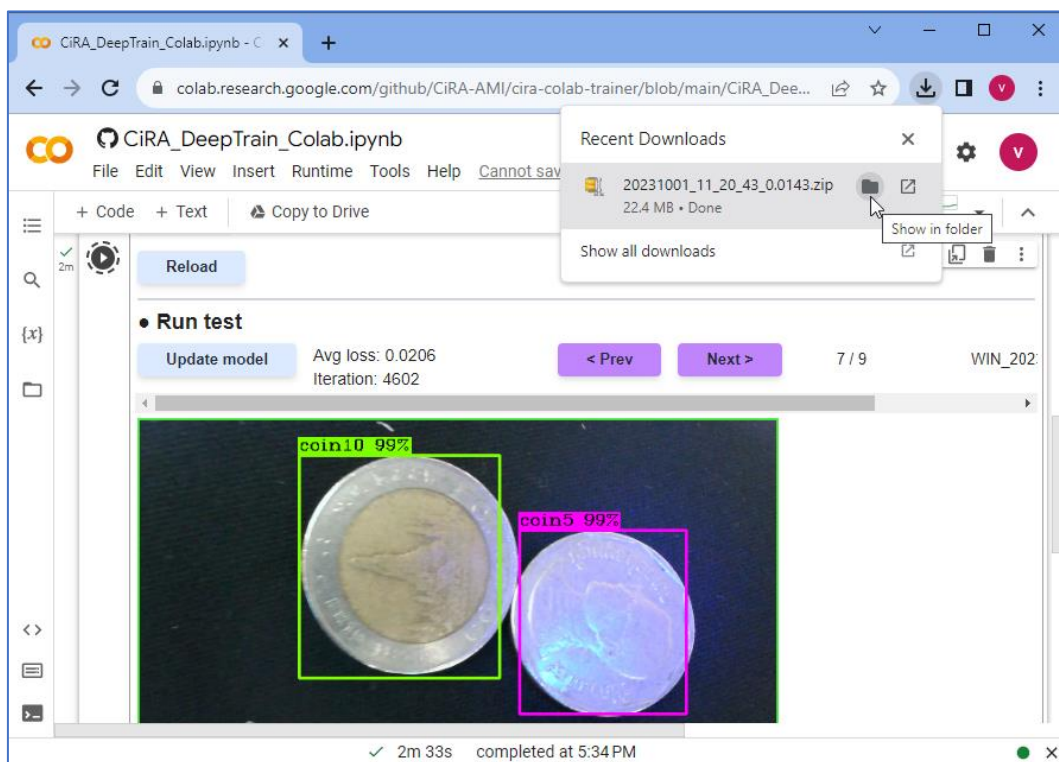
36. คลิก Update model เพื่อโหลดรูปภาพตัวอย่างมาแสดง



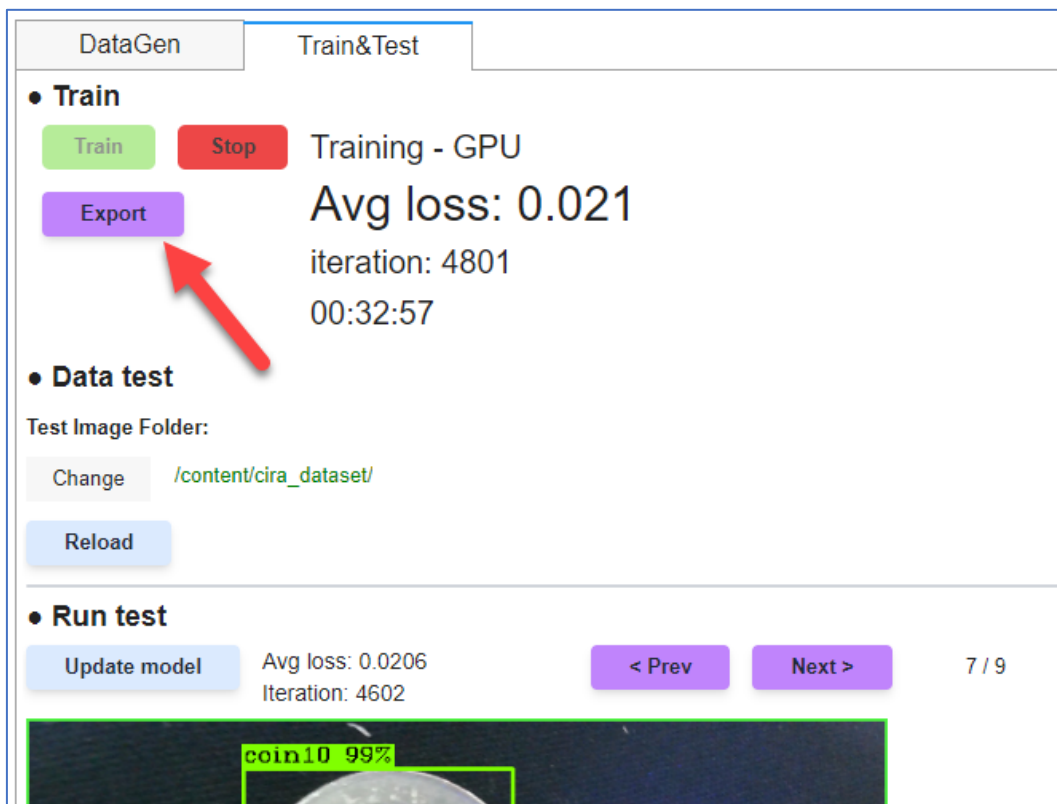
37. คลิกปุ่ม Next เพื่อเลื่อนดูตัวอย่างการตรวจจับเหรียญของแต่รูปภาพ



ผลลัพธ์ที่ถูกต้อง จะปรากฏดังรูป



38. เมื่อได้ผลลัพธ์ที่ถูกต้องแล้ว ให้คลิกปุ่ม Export

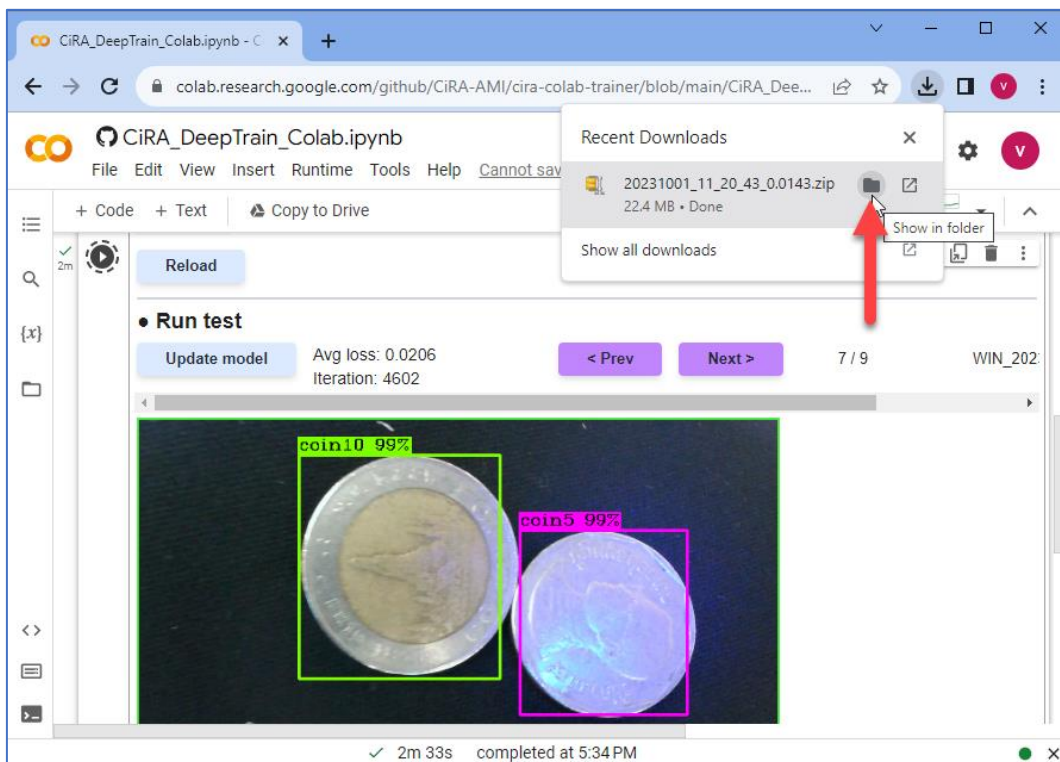


39. แถบคาน์โหลจะปรากฏขึ้นมา รอจนคาน์โหลครบ 100%

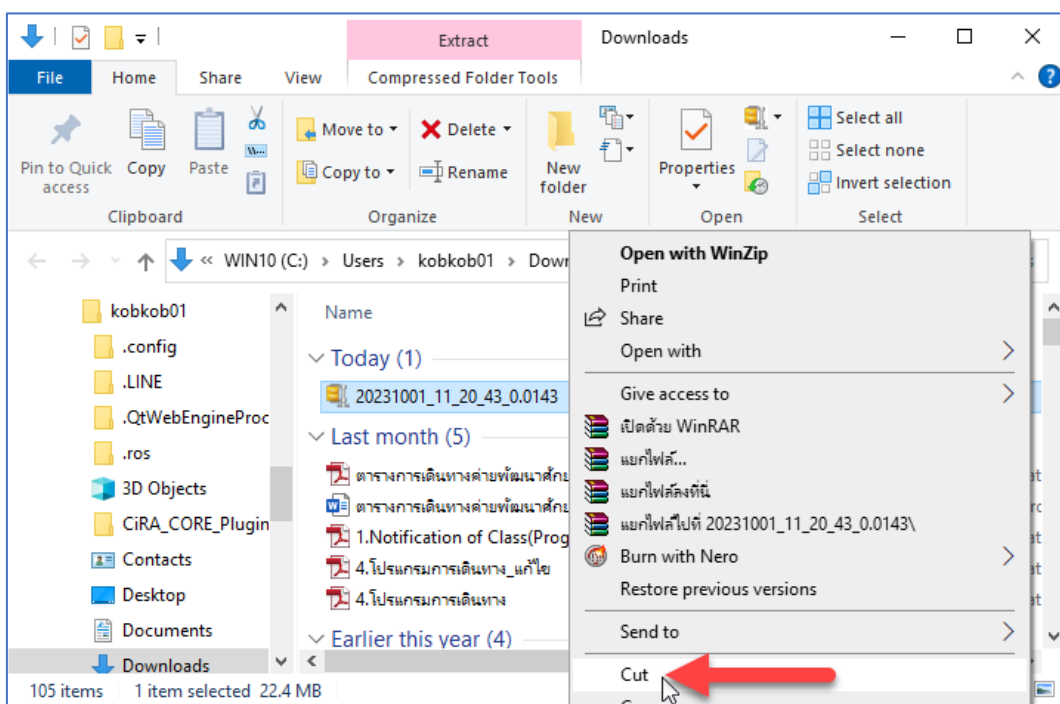




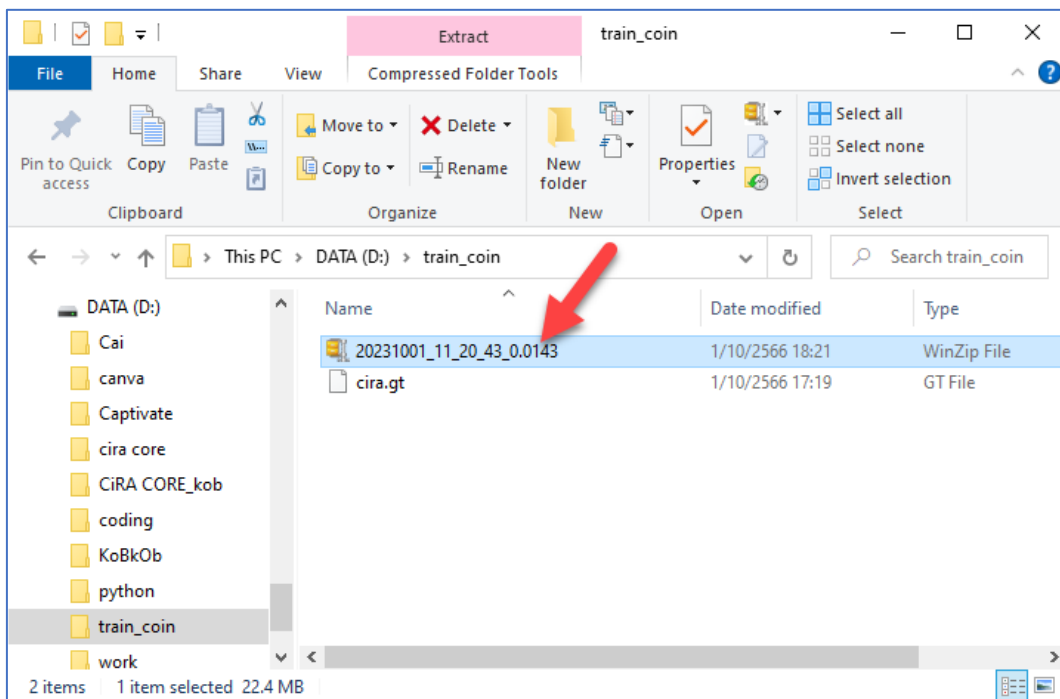
40. ระบบจะทำการดาวน์โหลดไฟล์ zip เข้ามาที่โฟลเดอร์ Download



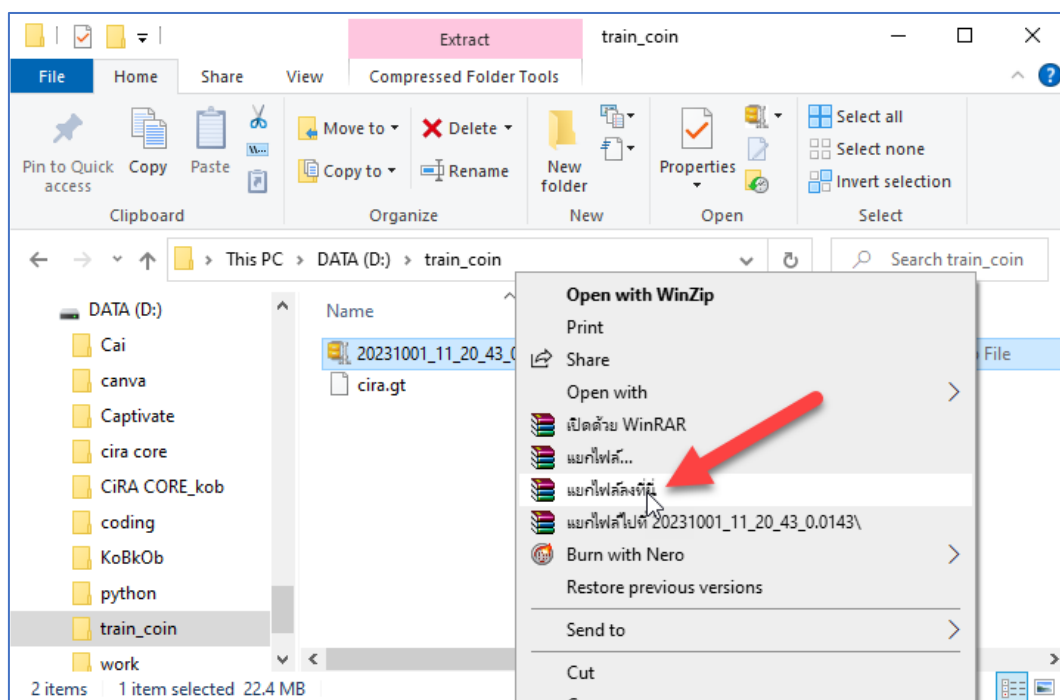
41. ทำการย้ายไฟล์ โดยคลิกขวาที่ไฟล์ zip เลือกคำสั่ง cut



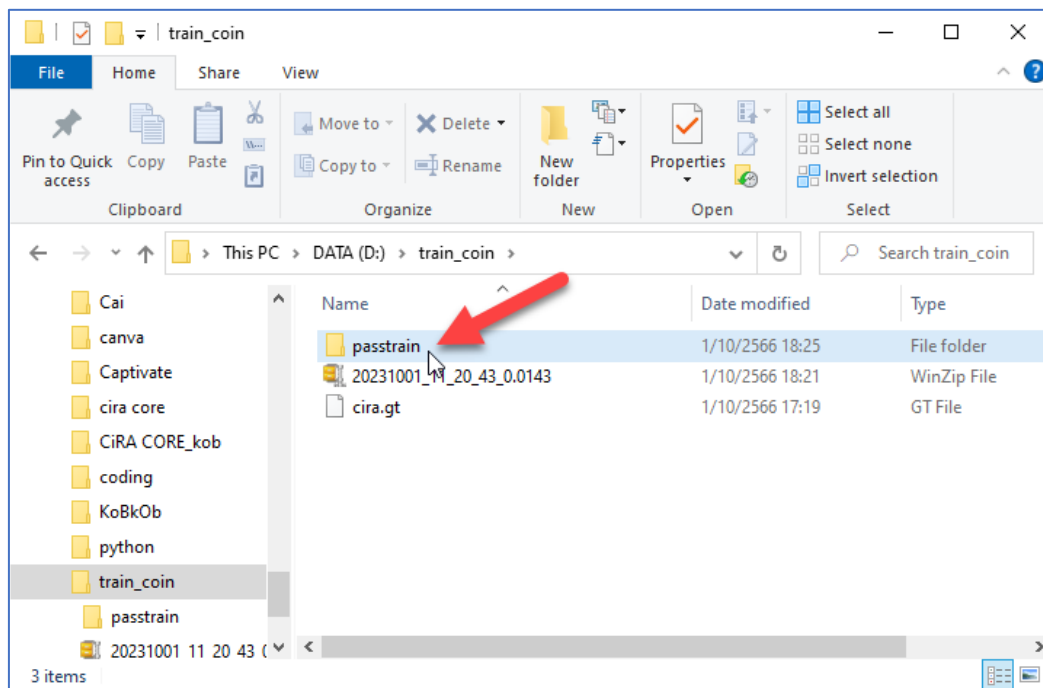
42. นำไฟล์ zip ไปวางในไดรฟ์ D:/



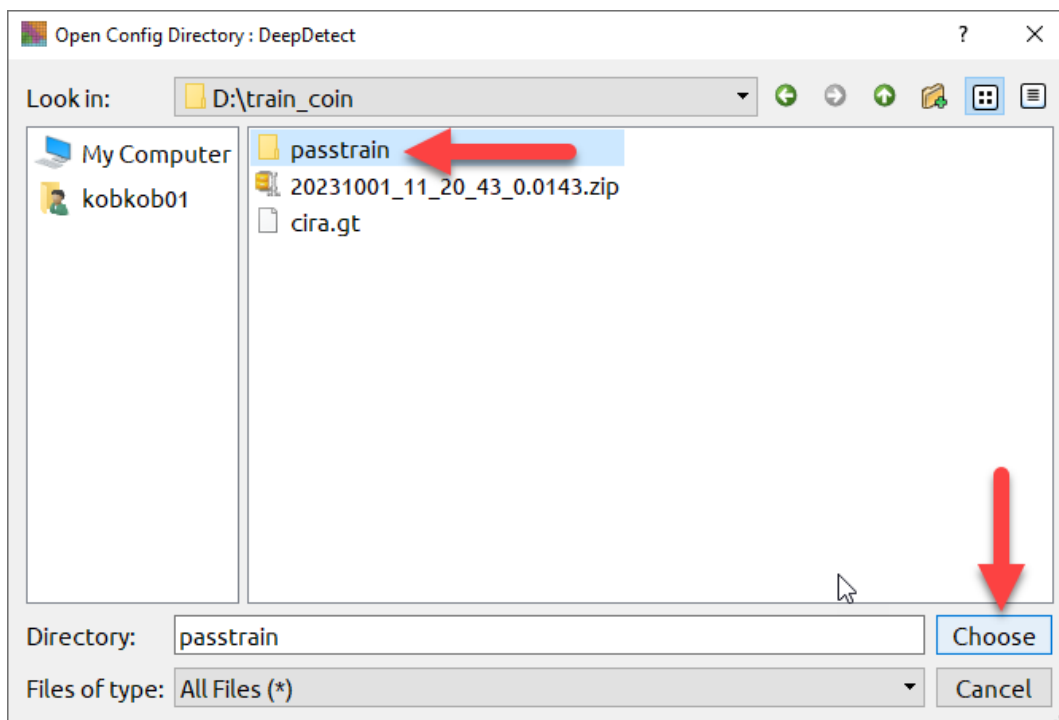
43. แยกไฟล์ โดยคลิกขวาที่ไฟล์ zip แล้วเลือกคำสั่ง แยกไฟล์ลงที่นี่

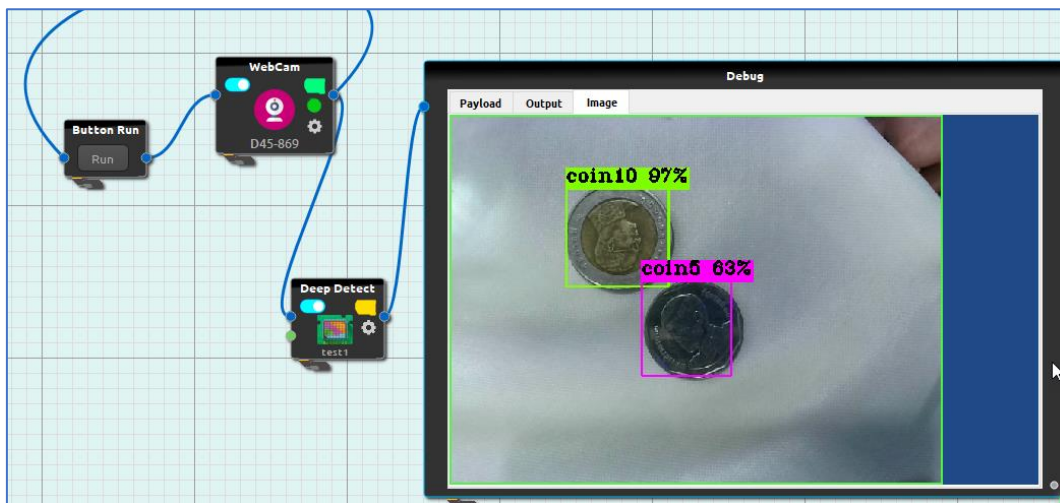


## 44. ทำการเปลี่ยนชื่อโฟลเดอร์



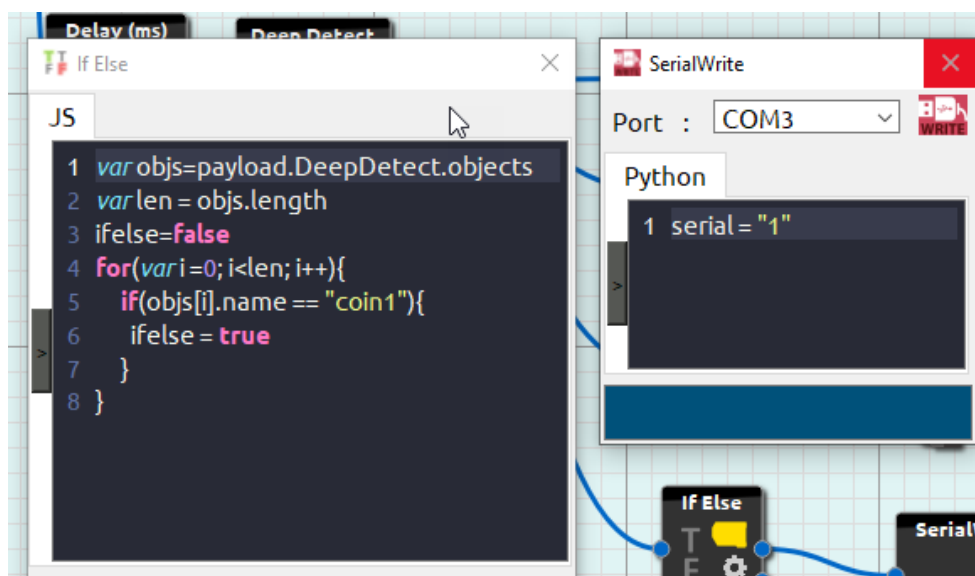
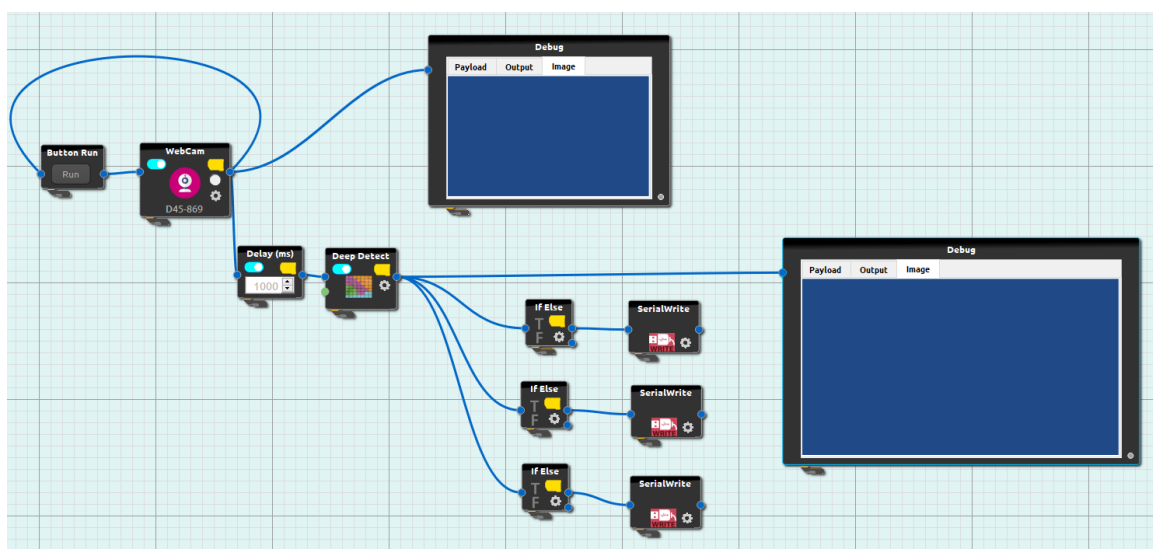
## 45. เปิด CIRA CORE ในส่วนของบล็อกร DeepDetect ให้คลิกเลือกโฟลเดอร์ที่ผ่านการเทรนจาก Google Colab มาใช้งาน



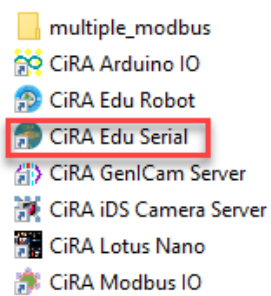


## การเชื่อมต่อกับบอร์ด Micro: bit

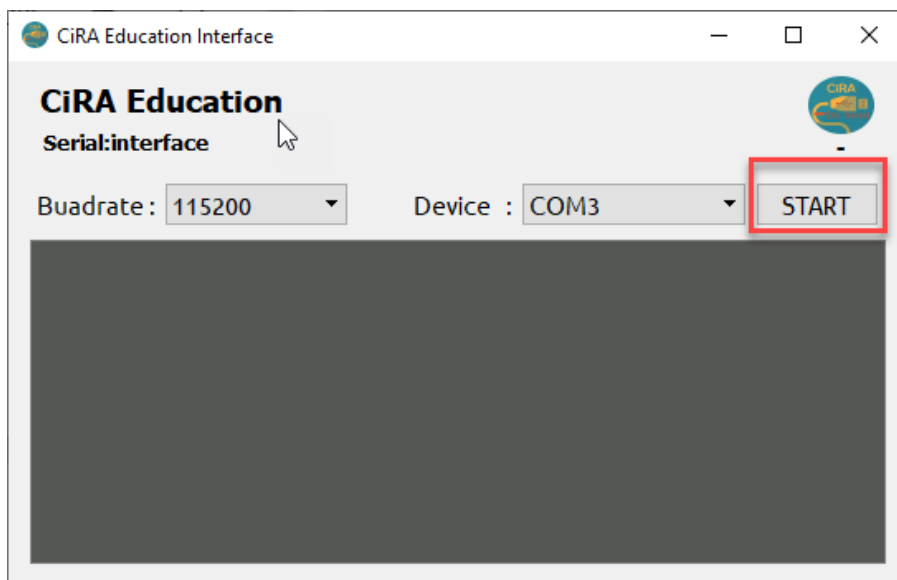
ใน Cira core ใส่บล็อกคำสั่งตามภาพ



การเชื่อมต่อ ให้ไปที่ C:\CiRA-CORE\shortcut\_module\CiRA Module ดับเบิลคลิก

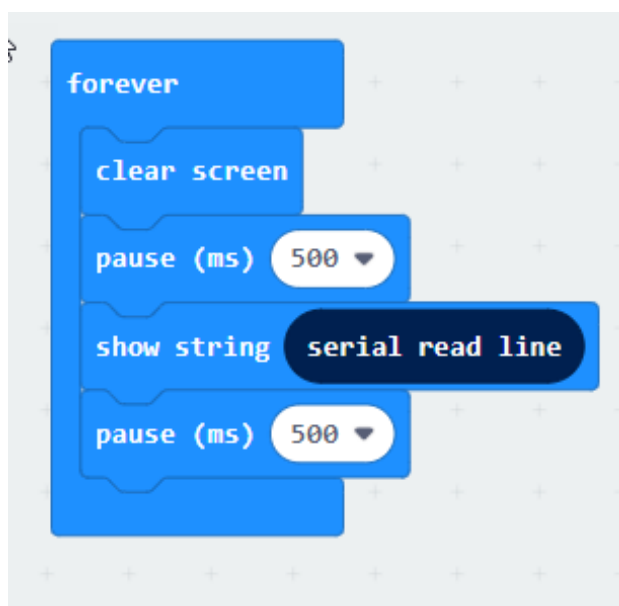


เสียบบอร์ด แล้วตรง Device มี port ขึ้นมา ให้กดปุ่ม start



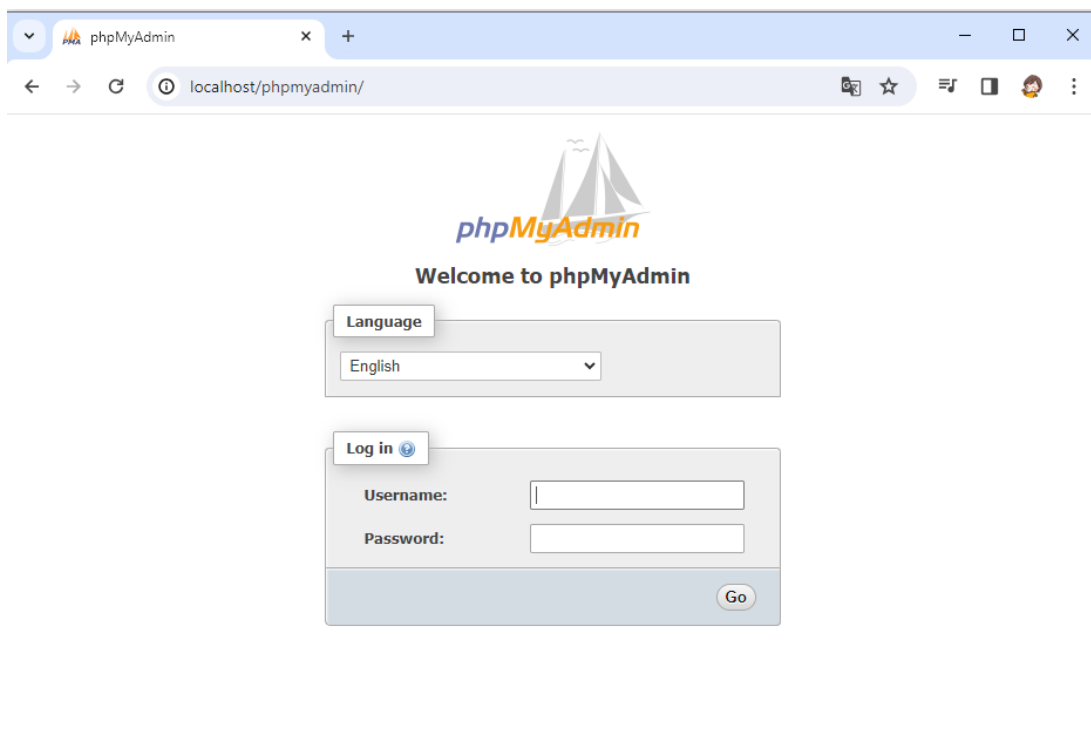
จากนั้นกลับไป Cira core

เปิด [makecord.microbit.org](http://makecord.microbit.org)

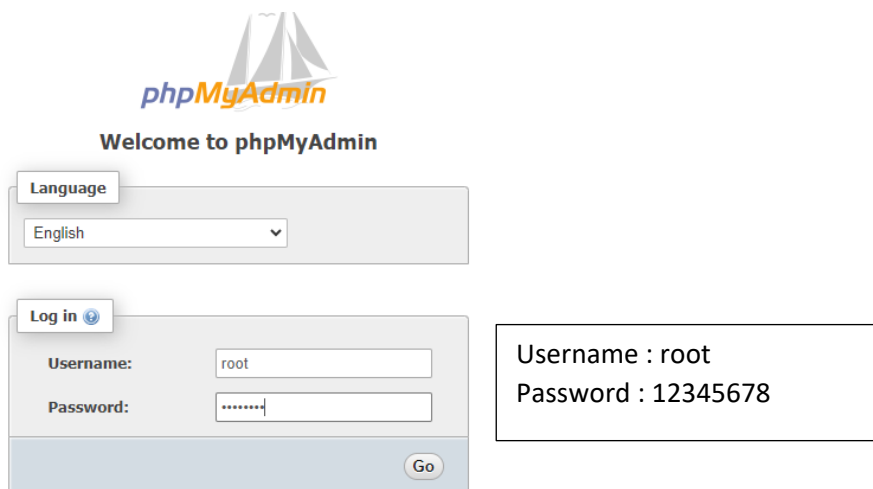


## Cira core ติดต่อกับฐานข้อมูล

1. เข้าที่ localhost โดยไปที่ Google chrome แล้วพิมพ์ว่า Localhost/phpMyAdmin

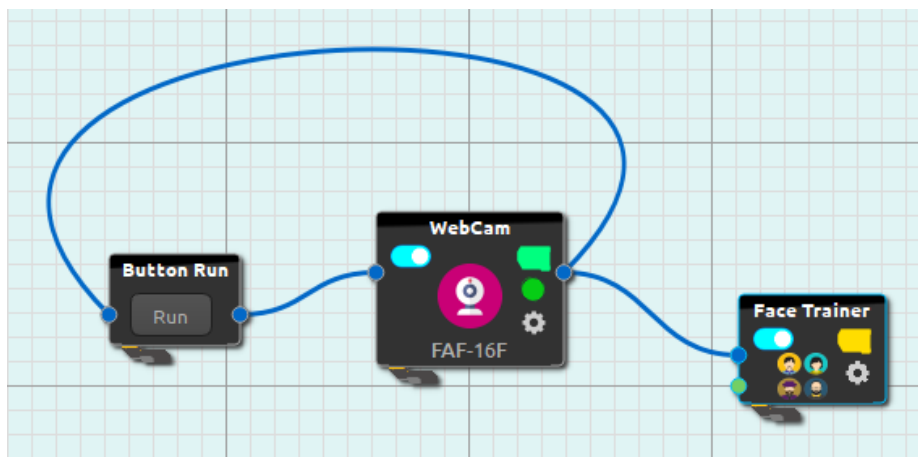


2. กรอก Username และ Password

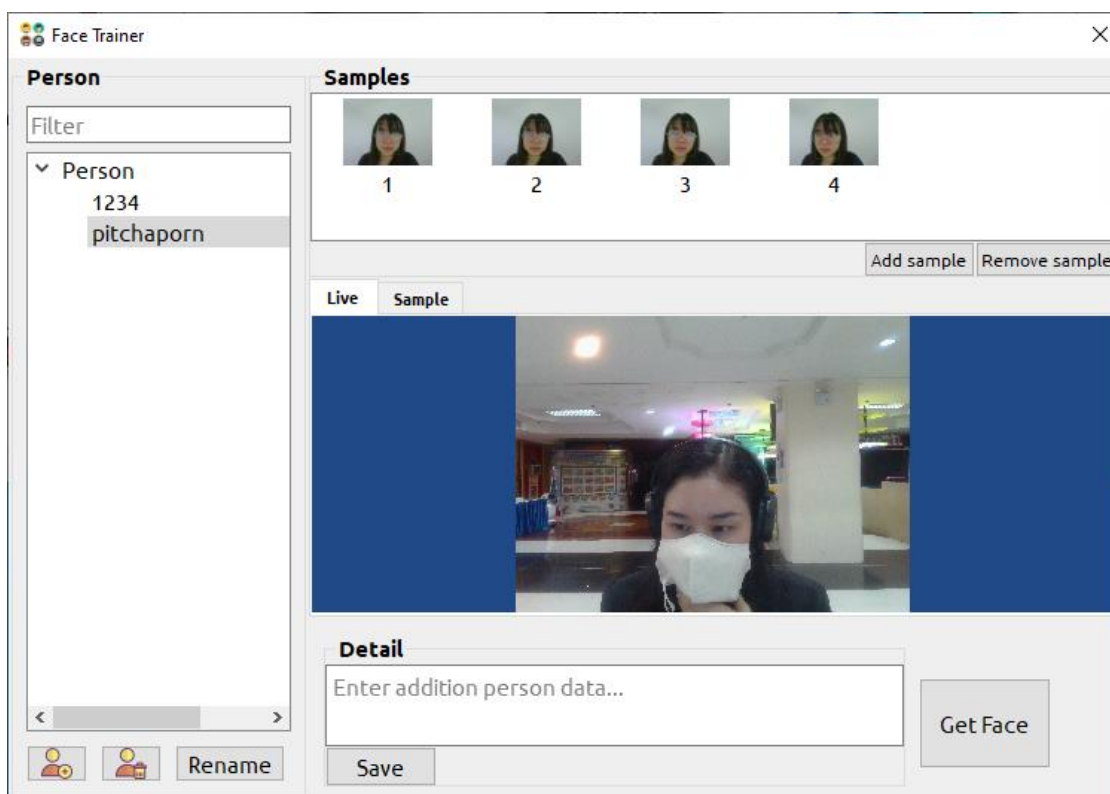


## ตรวจจับใบหน้า บันทึกลงฐานข้อมูล แจ้งเตือน line

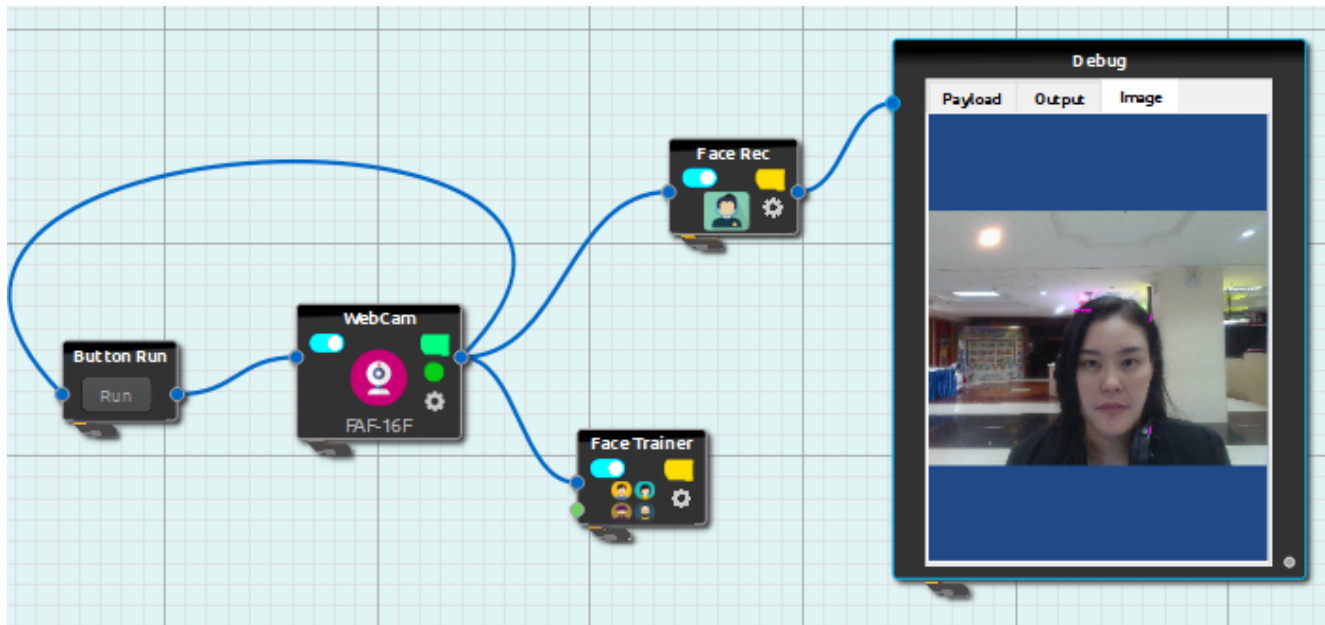
1. เปิด Cira core แล้วใส่คำสั่งดังนี้



ตั้งค่ากล้องให้เรียบร้อย แล้วตรง Face Trainer ให้ถ่ายรูปตัวอย่าง

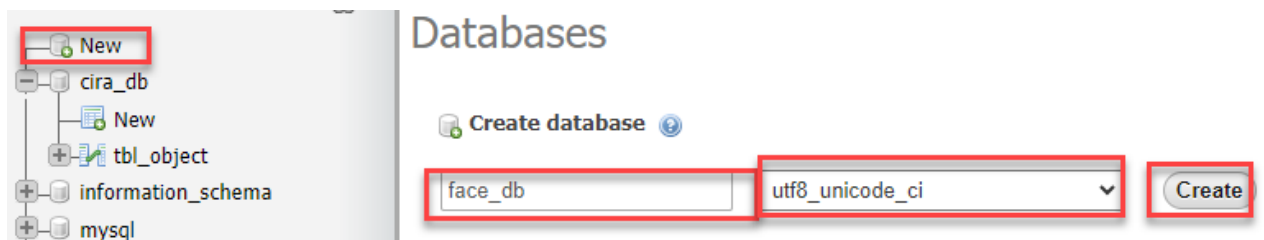


ใส่คำสั่งเพิ่ม



ไปที่ฐานข้อมูล

สร้างฐานข้อมูลใหม่



สร้างตาราง 2 ตาราง

ตารางที่ 1

Create table

Name:       Number of columns:

Name	Type	Length/Values	Default	Collation	Attributes	Null	Index
std_code	VARCHAR	6	None			<input type="checkbox"/>	PRIMARY <small>PRIMARY</small>
prefix	VARCHAR	10	None			<input type="checkbox"/>	---
name	VARCHAR	50	None			<input type="checkbox"/>	---
surname	VARCHAR	50	None			<input type="checkbox"/>	---
level	VARCHAR	1	None			<input type="checkbox"/>	---
room	VARCHAR	2	None			<input type="checkbox"/>	---



## ตารางที่ 2

The image shows two screenshots from phpMyAdmin. The top screenshot is the 'Create table' dialog where the table name is 'tbl\_checkin' and the number of columns is set to 4. The bottom screenshot is the 'Structure' tab for the table 'tbl\_checkin'. It shows a table with 4 columns: 'id\_check' (INT, PRIMARY), 'datetime' (DATETIME), 'std\_code' (VARCHAR(6)), and 'send\_line' (VARCHAR(1)). The 'id\_check' column is marked as the primary key. Below the table structure, there are fields for 'Table comments', 'Collation', and 'Storage Engine' (set to InnoDB). A 'PARTITION definition' section is also visible at the bottom.

กำหนด id\_check เป็น primary key

เพิ่มข้อมูลฐานข้อมูลก่อน

The image shows a screenshot of the phpMyAdmin interface. On the left sidebar, the database 'face\_db' is selected, and the table 'tbl\_students' is highlighted with a red box and a circled '1'. In the main area, the 'Structure' tab for 'tbl\_students' is open. The 'Insert' button in the top toolbar is highlighted with a red box and a circled '2'. The table structure table below has columns 'Column', 'Type', 'Function', 'Null', and 'Value'. The rows are: 'std\_code' (varchar(6)), 'prefix' (varchar(10)), 'name' (varchar(50)), 'surname' (varchar(50)), 'level' (varchar(1)), and 'room' (varchar(2)). A red box highlights the entire table structure area, and a circled '4' is placed over the 'Go' button at the bottom right of this area. Below the table structure, there is an 'Ignore' checkbox which is checked, and another table structure table with the same columns as above.

Server: localhost » Database: face\_db » Table: tbl\_students

[Browse](#)
[Structure](#)
[SQL](#)
[Search](#)
[Insert](#)
[Export](#)
[Import](#)
[Privileges](#)

Column	Type	Function	Null	Value
std_code	varchar(6)			123456
prefix	varchar(10)			นางสาว
name	varchar(50)			จิรดา
surname	varchar(50)			ขุนแผน
level	varchar(1)			4
room	varchar(2)			2

[Go](#)

✓ 1 row inserted.

```
INSERT INTO `tbl_students` (`std_code`, `prefix`, `name`, `surname`, `level`, `room`) VALUES ('123456', 'นางสาว', 'จิรดา', 'ขุนแผน', '4', '2');
```

[\[Edit inline\]](#) [\[Edit\]](#) [\[Create PHP code\]](#)

Run SQL query/queries on table face\_db.tbl\_students:

```
1 INSERT INTO `tbl_students` (`std_code`, `prefix`, `name`, `surname`, `level`, `room`) VALUES ('123456', 'นางสาว', 'จิรดา', 'ขุนแผน', '4', '2');
```

Columns

```
std_code
prefix
name
surname
level
room
```

[SELECT \\*](#)
[SELECT](#)
[INSERT](#)
[UPDATE](#)
[DELETE](#)
[Clear](#)
[Format](#)

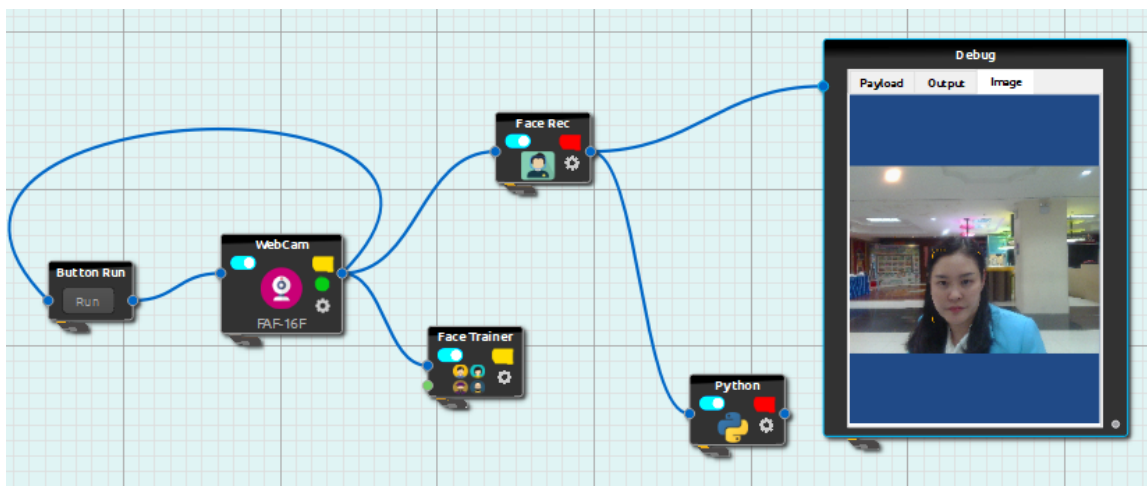
[Get auto-saved query](#)

Bind parameters

[ Delimiter : ]  Show this query here again  Retain query box  Rollback when finished  Enable foreign key checks [Go](#)

เสร็จแล้วกลับมาที่ cira code

ใส่ไฟทอน



## ใส่โค้ด

Python Version 2. Python311 Stop

comment:  Save Load

Python

```

1 import mysql.connector
2 import requests
3 import datetime
4
5 def sendline(msg):
6     url = 'https://notify-api.line.me/api/notify'
7     token = 'NiG1BCJ1tn8Ak2qe17fo1G8cpRnZ7zWT8qBbGBneUtx'
8     headers = {'content-type': 'application/x-www-form-urlencoded', 'Authorization': 'Bearer '+token}
9
10    r = requests.post(url, headers=headers, data = {'message':msg})
11    return r.ok
12
13    mydb = mysql.connector.connect (
14        host = "localhost",
15        user = "root",
16        password = "12345678",
17        database = "face_db"
18    )
19
20    mycursor = mydb.cursor()
21
22    dd = datetime.datetime.now()
23
24    objects = payload["FaceRec"]["face_array"]
25    face_count = payload["FaceRec"]["count"]
26
27    if (face_count <=0):
28        print("ไม่พบข้อมูลใบหน้า")
29        payload["show_message"] = "ไม่พบข้อมูลใบหน้า"
30    else :
31        for objs in objects :
32            sql = "select * from tbl_students where std_code = %s"
33            val = (objs["name"],)
34            mycursor.execute(sql,val)
35            records = mycursor.fetchall()
36            if (mycursor.rowcount <= 0):
37                print("ไม่พบข้อมูล ",objs["name"],"ในฐานข้อมูล")
38            else :
39                std_name = records[0][1]+records[0][2]+" "+records[0][3]
40                sql = "select datetime, send_line from tbl_checkin where std_code = %s and DATE(datetime) = %s "
41                val = (objs["name"], dd.strftime("%Y-%m-%d"),)
42                mycursor.execute(sql,val)
43                records = mycursor.fetchall()
44                if (mycursor.rowcount <=0):
45                    sql = "insert into tbl_checkin (check_id, datetime, std_code, send_line) values(NULL, %s, %s, '0')"
46                    val = (dd.strftime("%Y-%m-%d %H:%M"), objs["name"],)
47                    mycursor.execute(sql,val)
48                    mydb.commit()
49
50
51                msg = "บันทึกข้อมูลลงเวลา ของ "+std_name+" เมื่อ "+dd.strftime("%Y-%m-%d %H:%M")
52                sl = sendline(msg)
53
54                if(sl):
55                    sql = "update tbl_checkin set send_line = '1' where std_code = %s"
56                    val = (objs["name"],)
57                    mycursor.execute(sql,val)
58                    mydb.commit()
59

```

Server: localhost » Database: face\_db » Table: tbl\_checkin

Showing rows 0 - 0 (1 total, Query took 0.0007 seconds.)

```
SELECT * FROM `tbl_checkin`
```

Number of rows: 25 Filter rows: Search this table

	check_id	datetime	std_code	send_line
<input type="checkbox"/> Edit Copy Delete	3	2024-04-02 22:24:00	654321	1

Number of rows: 25 Filter rows: Search this table

Query results operations: Print Copy to clipboard Export Display chart Create view

```
python
import mysql.connector
import requests
import datetime
```

- นำเข้า (import) โมดูล `mysql.connector` สำหรับการเชื่อมต่อกับฐานข้อมูล MySQL, โมดูล `requests` สำหรับการส่ง HTTP requests, และโมดูล `datetime` สำหรับการจัดการวันที่และเวลา.

```
python
def sendline(msg):
```

- นิยามฟังก์ชัน `sendline` ที่มีพารามิเตอร์ `msg` เพื่อใช้ส่งข้อความไปยัง LINE Notify.

```
python
url = 'https://notify-api.line.me/api/notify'
```

- กำหนด URL ของ LINE Notify API.

```
python
token = 'udNv6szmofMMu1IoeAMAd...mLbE9Cy6gfDp4gRTTNF'
```

- กำหนด token สำหรับการอนุญาตการเข้าถึง LINE Notify.

```
python Copy code
headers = {'content-type': 'application/x-www-form-urlencoded', 'Authorization': 'Be
```

- กำหนด headers สำหรับ HTTP request โดยใส่ token ที่ได้รับการอนุญาต.

```
python Copy code
r = requests.post(url, headers=headers, data = {'message':msg})
```

- ส่ง POST request ไปยัง LINE Notify API พร้อมข้อมูลข้อความที่จะส่ง.

```
python Copy code
return r.ok
```

- คืนค่า boolean จากการตอบกลับของ request ว่าสำเร็จหรือไม่.

```
python Copy code
mydb = mysql.connector.connect (
```

- เริ่มการเชื่อมต่อกับฐานข้อมูล MySQL.

```
python Copy code
host = "localhost",
user = "root",
password = "",
database = "face_db"
)
```

- กำหนดค่าการเชื่อมต่อฐานข้อมูล ได้แก่ โฮสต์, ชื่อผู้ใช้, รหัสผ่าน (ในที่นี้ไม่มีรหัสผ่าน), และชื่อฐานข้อมูล.

```
python Copy code
mycursor = mydb.cursor()
```

- สร้าง cursor สำหรับการดำเนินการกับฐานข้อมูล.

```
python Copy code
dd = datetime.datetime.now()
```

- กำหนดตัวแปร `dd` ให้เป็นวันที่และเวลาปัจจุบัน.

ต่อไปนี้เป็นส่วนที่จัดการกับข้อมูลการจดจำใบหน้าและบันทึกการเช็คคิน:

```
python Copy code
objects = payload["FaceRec"]["face_array"]
face_count = payload["FaceRec"]["count"]
```

- ดึงข้อมูลจาก payload ซึ่งรวมถึงอาร์เรย์ของใบหน้าที่ยัดเข้าได้และจำนวนของใบหน้า.

```
python Copy code
if (face_count <=0) :
```

- ตรวจสอบว่าไม่มีใบหน้าใดๆ ถูกจัดเข้าหรือไม่.

```
python Copy code
print("ไม่พบข้อมูลใบหน้า")
payload["show_message"] = "ไม่พบข้อมูลใบหน้า"
```

- ถ้าไม่พบใบหน้า, แสดงและบันทึกข้อความว่าไม่พบข้อมูลใบหน้า.

```
python Copy code
else :
```

- ถ้าพบใบหน้า:

```
python Copy code
for objs in objects :
```

- วนลูปตามจำนวนใบหน้าที่ยัดเข้าได้.

```
python Copy code
sql = "select * from tbl_students where std_code = %s"
```



- กำหนดคำสั่ง SQL เพื่อค้นหาข้อมูลนักเรียนจากฐานข้อมูลตามรหัสที่ระบุ.

```
python Copy code  
  
val = (objs["name"], )
```

- กำหนดค่าพารามิเตอร์สำหรับคำสั่ง SQL.

```
python Copy code  
  
mycursor.execute(sql, val)
```

- ประมวลผลคำสั่ง SQL ด้วยพารามิเตอร์ที่กำหนด.

```
python Copy code  
  
records = mycursor.fetchall()
```

- ดึงข้อมูลทั้งหมดที่คืนมาจากคำสั่ง SQL.

```
python Copy code  
  
if (mycursor.rowcount <= 0) :
```

- ตรวจสอบว่าไม่มีระเบียบข้อมูลที่ตรงกับคำสั่ง SQL หรือไม่.

```
python Copy code  
  
print("ไม่พบข้อมูล ", objs["name"], "ในฐานข้อมูล")
```

- ถ้าไม่พบข้อมูล, แสดงข้อความว่าไม่พบข้อมูลนั้นๆ ในฐานข้อมูล.

```
python Copy code
else :
```

- ถ้าพบข้อมูล:

```
python Copy code
std_name = records[0][1]+records[0][2]+" "+records[0][3]
```

- กำหนดชื่อนักเรียนจากข้อมูลที่คืนมา.

```
python Copy code
sql = "select datetime, send_line from tbl_checkin where std_code = %s and"
```

- กำหนดคำสั่ง SQL เพื่อตรวจสอบว่ามีการเช็คอินของนักเรียนในวันนั้นแล้วหรือไม่.

```
python Copy code
val = (objs["name"], dd.strftime("%Y-%m-%d"), )
```

- กำหนดค่าพารามิเตอร์สำหรับคำสั่ง SQL, โดยใช้วันที่ปัจจุบัน.

```
python Copy code
mycursor.execute(sql, val)
```



- ประมวลผลคำสั่ง SQL.

```
python Copy code
records = mycursor.fetchall()
```

- ดึงข้อมูลทั้งหมดที่คืนมาจากคำสั่ง SQL.

```
python Copy code
if (mycursor.rowcount <=0):
```

- ตรวจสอบว่าไม่มีระเบียบการเช็คอินในวันนั้น.

```
python Copy code
sql = "insert into tbl_checkin (check_id, datetime, std_code, send_li
```

- กำหนดคำสั่ง SQL เพื่อบันทึกการเช็คอินใหม่.

```
python Copy code
val = (dd.strftime("%Y-%m-%d %H:%M"), objs["name"], )
```

- กำหนดค่าพารามิเตอร์สำหรับคำสั่ง SQL ใหม่, รวมถึงวันที่และเวลาปัจจุบัน.

```
python Copy code
mycursor.execute(sql, val)
mydb.commit() 
```

- ประมวลผลคำสั่ง SQL และยืนยันการเปลี่ยนแปลงในฐานข้อมูล.

```
python Copy code
msg = "บันทึกข้อมูลลงเวลา ของ "+std_name+" เมื่อ "+dd.strftime("%Y-%m-%d %H:
```

- สร้างข้อความสำหรับส่งไปยัง LINE Notify.

```
python Copy code
s1 = sendline(msg)
```

- ส่งข้อความไปยัง LINE Notify และรับค่า boolean ที่บ่งบอกว่าการส่งสำเร็จหรือไม่.

```
python Copy code
if(s1) :
```

- ถ้าการส่งสำเร็จ:

```
python Copy code
sql = "update tbl_checkin set send_line = '1' where std_code = %s"
```

- กำหนดคำสั่ง SQL เพื่ออัปเดตสถานะการส่งข้อความในฐานข้อมูล.

```
python Copy code
val = (objs["r""], )
```

- กำหนดค่าพารามิเตอร์สำหรับคำสั่ง SQL นี้.

```
python Copy code
mycursor.execute(sql, val)
mydb.commit()
```

- ประมวลผลคำสั่ง SQL และยืนยันการเปลี่ยนแปลงในฐานข้อมูล.

โค้ดนี้แสดงการใช้งาน MySQL, การจัดการวันที่, การส่ง HTTP requests และการควบคุมโฟลว์ของโปรแกรมด้วยเงื่อนไข. ควรจะมีการจัดการข้อผิดพลาดและความปลอดภัยของข้อมูลเพิ่มเติม, เช่น การซ่อน token สำหรับ LINE Notify และการใช้ placeholders อย่างเหมาะสมในคำสั่ง SQL เพื่อป้องกัน SQL Injection.