

HUB8735



RTSP
Real-Time Streaming Protocol



國產IC應用實作開發系列

HUB 8735 智慧儀表工業應用

(HUB 8735 Detection Meter)

講師: 章育銘

112/08/24



物聯網智造基地

I O T S E R V I C E H U B



網址:

<https://github.com/wildman8606/HUB8735-Detection-Meter>



20230519 [國產IC開發方案]HUB 8735實務Training:
<https://www.youtube.com/watch?v=YZafxlf89aA>

講師介紹

講師: 章育銘

- # 退役競賽獎金獵人
- # 台灣門薩 選委會
- # 資訊工業策進會 講師
- # 工業局物聯網策略解決專家
- # 工業技術研究院 副工程師
- # 甲種電匠
- # 交大電控博士休學中
- ... +99



講師介紹-相關教學

講師: 章育銘

Coscup 2023 - Open Edge AI & TinyML - 手把手帶領多款國產Smart AI CAM與語音手勢辨識開發板:

https://www.youtube.com/watch?v=l2_6VLrZbk8&t=598s

20230519 [國產IC開發方案]HUB 8735實務Training:

<https://www.youtube.com/watch?v=YZafxlf89aA>

[十分鐘快速上手AI Camera] 國產瑞昱AMB82 Mini(Realtek Ameba Pro2)人臉偵測範例:

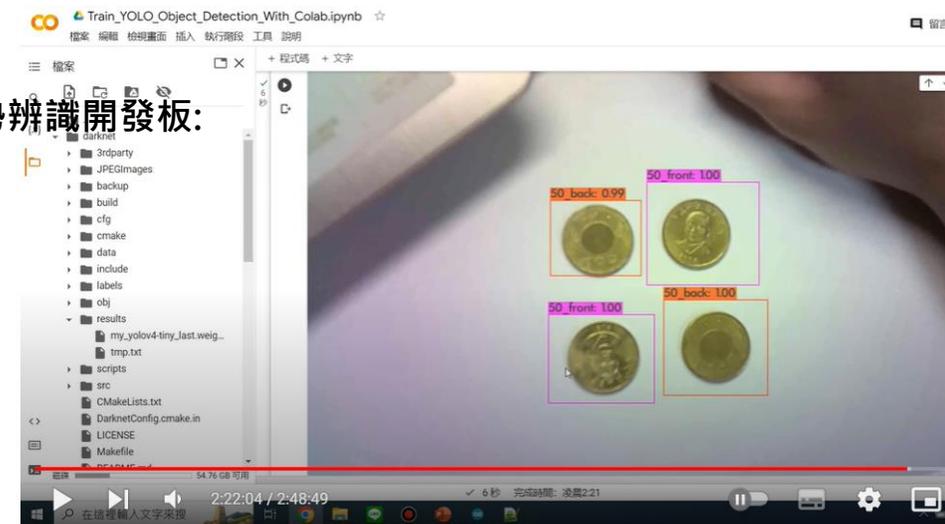
<https://www.youtube.com/watch?v=RGyOvpVo49M>

WE-I Plus智慧儀錶辨識:

https://www.ideas-hatch.com/evb_share_detail.jsp?id=70

CoreMaker-01 串接 Arduino 環境AI 語音辨識&手勢判斷 教學:

https://www.youtube.com/watch?v=dMBI8v92u_4&t=1050s



20230519 [國產IC開發方案]HUB 8735實務Training(講師-章育銘)



三、訓練數據蒐集

下載ARC GNU Toolchain

Step1. 點選或git clone 下載套件壓縮檔案

```
https://github.com/foss-for-synopsys-dwc-arc-processors/toolchain/releases/download/arc_2020.09_release/arc_gnu_2020.09_prebuilt_elf32_le_linux_install.tar.gz
```

Step2. 解壓縮檔案

```
$ sudo tar xzvf arc_gnu_2020.09_prebuilt_elf32_le_linux_install.tar.gz
```

Step3. 更改環境變數

```
$ gedit ~/.bashrc
```

將下列加入檔案之中，並儲存

```
export PATH=/home/miku/arc_gnu_2020.09_prebuilt_elf32_le_linux_install/bin:$PATH
```

```
$ source ~/.bashrc
```

圖 1. Gedit Editing Tools

2022 © 資訊工業策進會 Institute for Information Industry



1. 前言
2. 以Arduino IDE操作Smart-CAM
3. labellmg標注工具介紹
4. 使用Google Colab製作AI模型
5. 將模型燒入晶片
6. 其他

HUB 8735

 購買通路

 智造工具包

晶片原廠
瑞昱半導體

晶片採用
Ameba RTL8735

【開發板特點】

- 兼容Arduino開發特性
- 具備多功能影像處理的高度集成模組
- 內置NPU AI 運算引擎加速處理AI模型
- 802.11 a/b/g/n 雙頻Wi-Fi與BLE低耗電藍牙傳輸
- 可廣泛應用於各種結合影像識別或AI運算之物聯網場域



經濟部工業局廣告

前言



圖 1. 常見電表



圖 2. HUB8735



圖 3. 瓦斯錶



圖 4. 水錶

前言

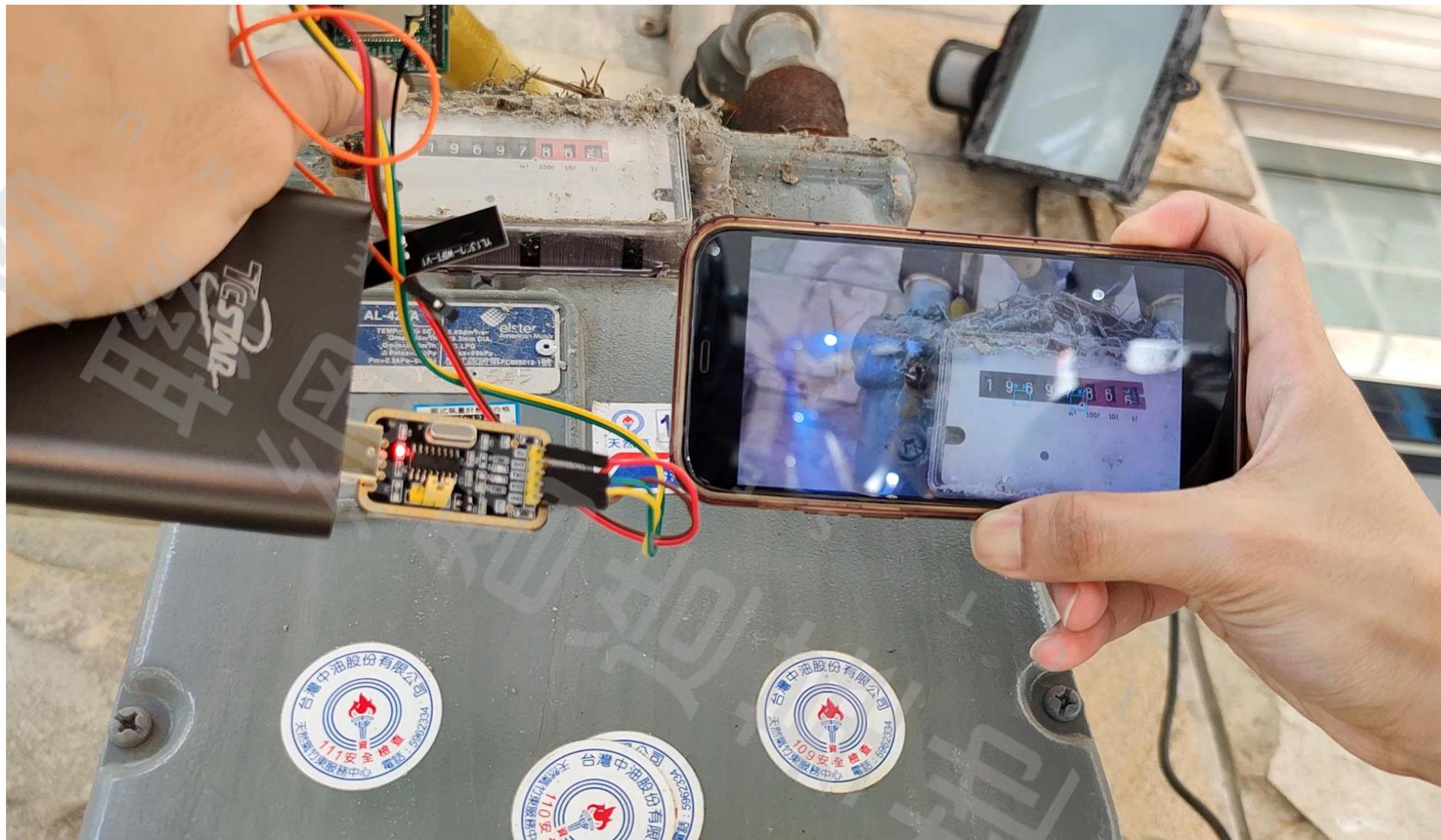


影1. HUB8735智慧水表辨識

前言



影1. HUB8735智慧瓦斯表辨識



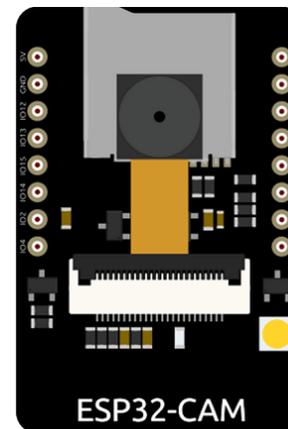
影2. HUB8735智慧瓦斯表辨識

HUB 8735 Smart AI CAM



- SPI x 1組
- I2C x 2組
- PWM x 4組
- UART x 2組
- ADC x 7組
- GPIO x 15組

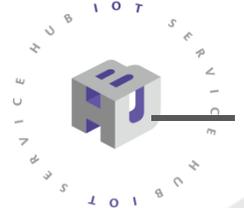
ESP32-CAM



- SPI x 1組
- I2C x 0組
- PWM x 3組
- UART x 1組
- ADC x 7組
- GPIO x 10組

✓ HUB 8735擁有與ESP32-CAM相同的PIN腳定義與排序，方便硬體互換

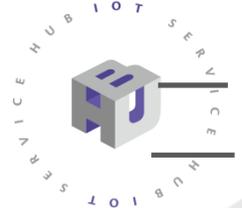
✓ HUB 8735使用相同於ESP32-CAM的軟體指令集，不必再重新熟悉軟體開發架構



前言

表1. HUB 8735 Smart AI CAM

功能	描述
處理器	RTL8735B AIOT國產晶片
影像輸入	搭配國產Full HD 1080P CMOS感測
語音輸入	內建MIC語音輸入功能
儲存裝置	支援SD記憶卡
無線連通	Wi-Fi 2.4GHz/5GHz Bluetooth BLE 無線影像串流
影像壓縮	H.264/265
AI處理	提供多種pre-trained AI models供快速上手
UART介面	提供UART串接多種控制平台，如Arduino等，使用UART控制Smart AI CAM的行為
USB介面	USB影像輸出
I/O擴充板	依照開發者需求擴充功能。 Speaker語音輸出功能 IMU sensor 擴充溫度、震動、濕度等功能



以Arduino IDE操作Smart-CAM -HUB 8735環境準備

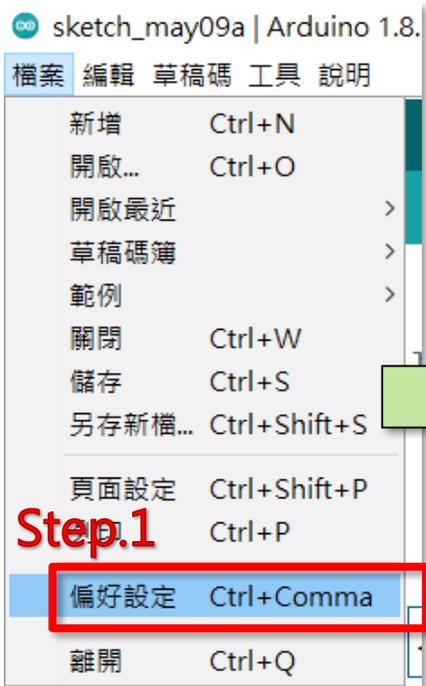


步驟1. 需安裝Arduino IDE 1.8.19之後的版本

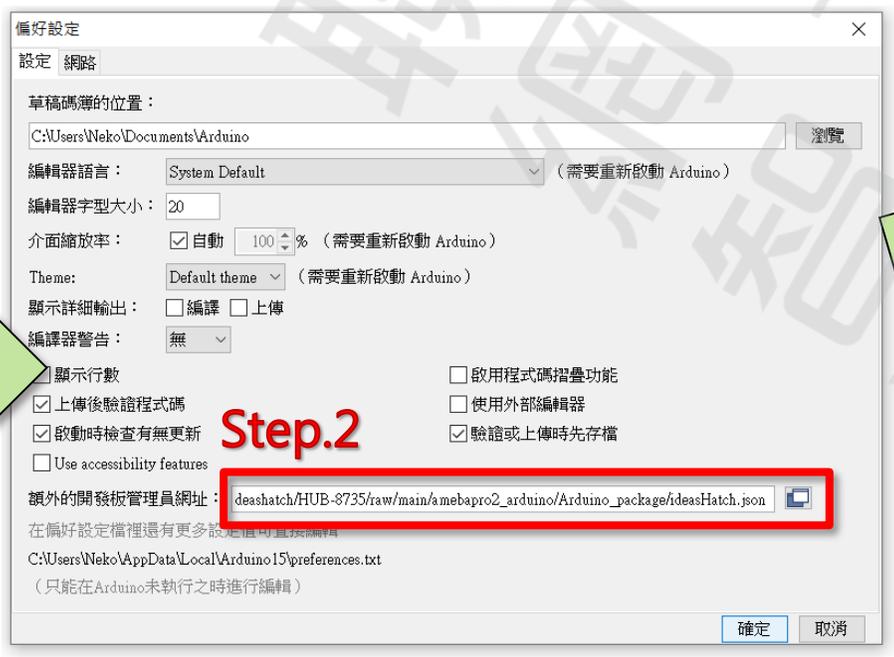
步驟2. 開啟Arduino IDE，透過檔案->偏好設定，在Additional Boards Manager URLs 填入以下連結後：

https://github.com/ideashatch/HUB-8735/raw/main/amebapro2_arduino/Arduino_package/ideasHatch.json

步驟3. 再從工具->開發板管理員中找到HUB 8735的開發板資料。



Step.1



Step.2



Step.3

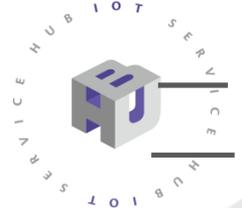


Step.4

圖1. 進入'偏好設定'

圖2. 丟入連結

圖4. 輸入'8735'找到開發板安裝



以Arduino IDE操作Smart-CAM - 燒錄HUB 8735 (1)

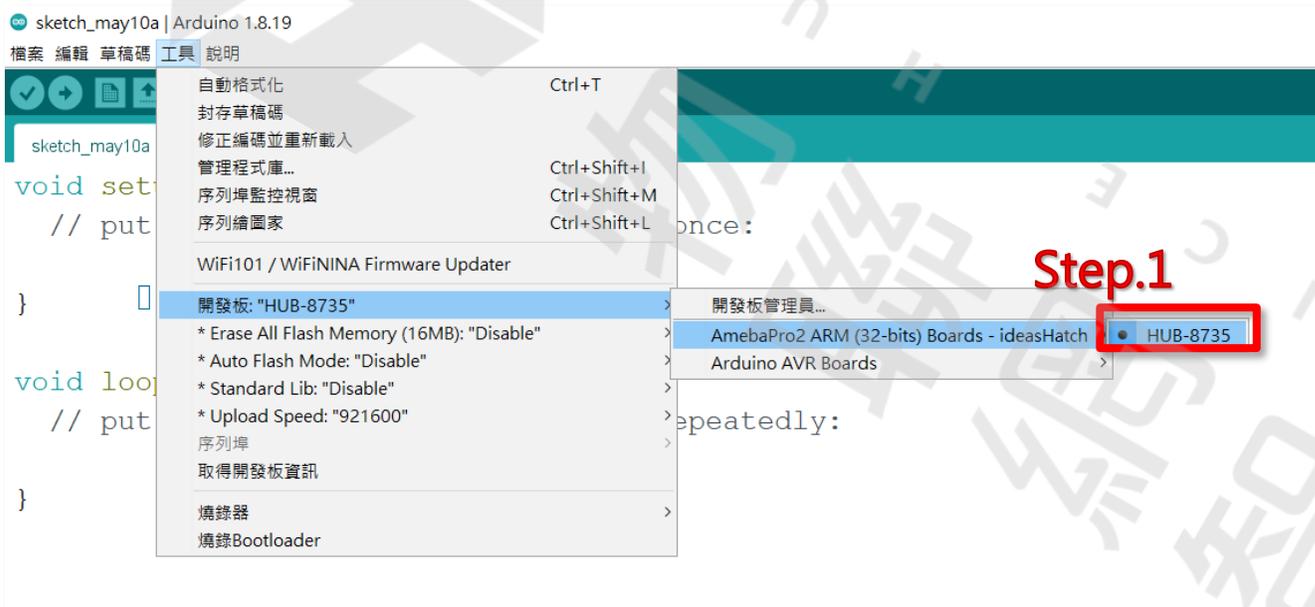


圖1.選擇'HUB-8735'開發板設定

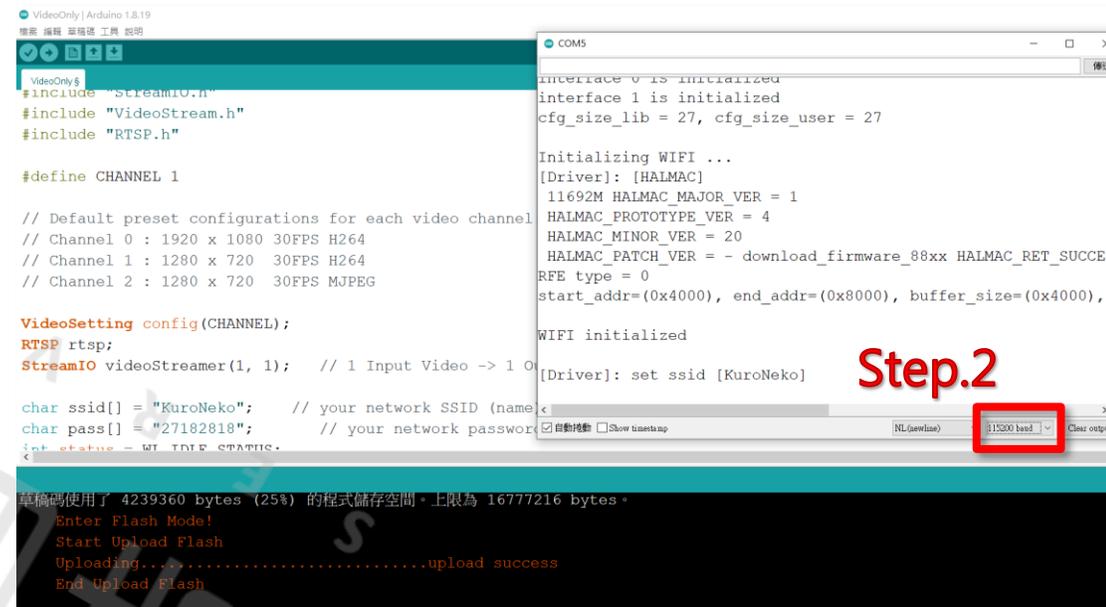
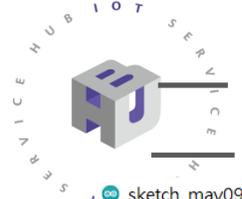


圖2.燒入範例後選擇'115200'鮑率(baud rate)



以Arduino IDE操作Smart-CAM - 燒錄HUB 8735 (2)

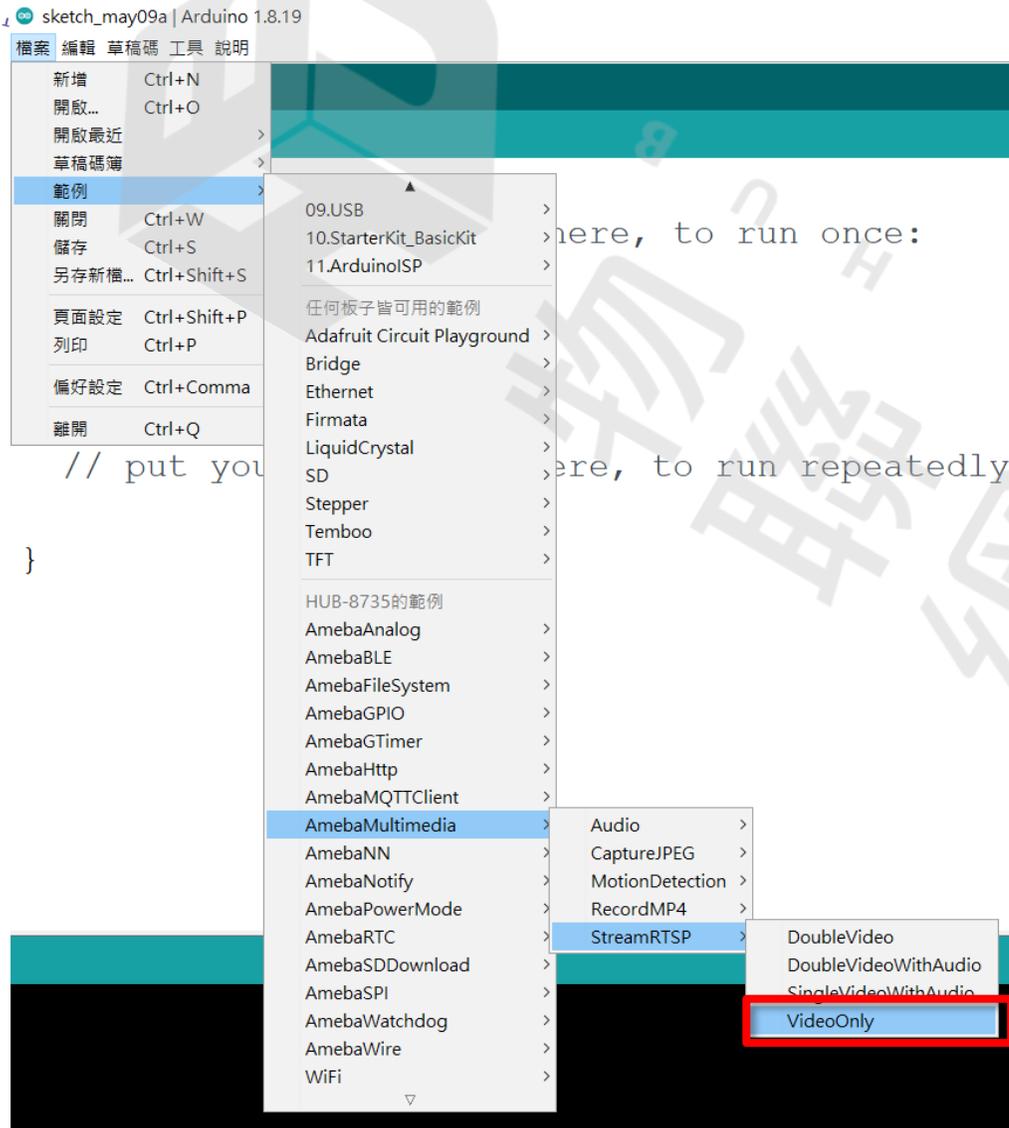


圖1. 範例'VideoOnly'

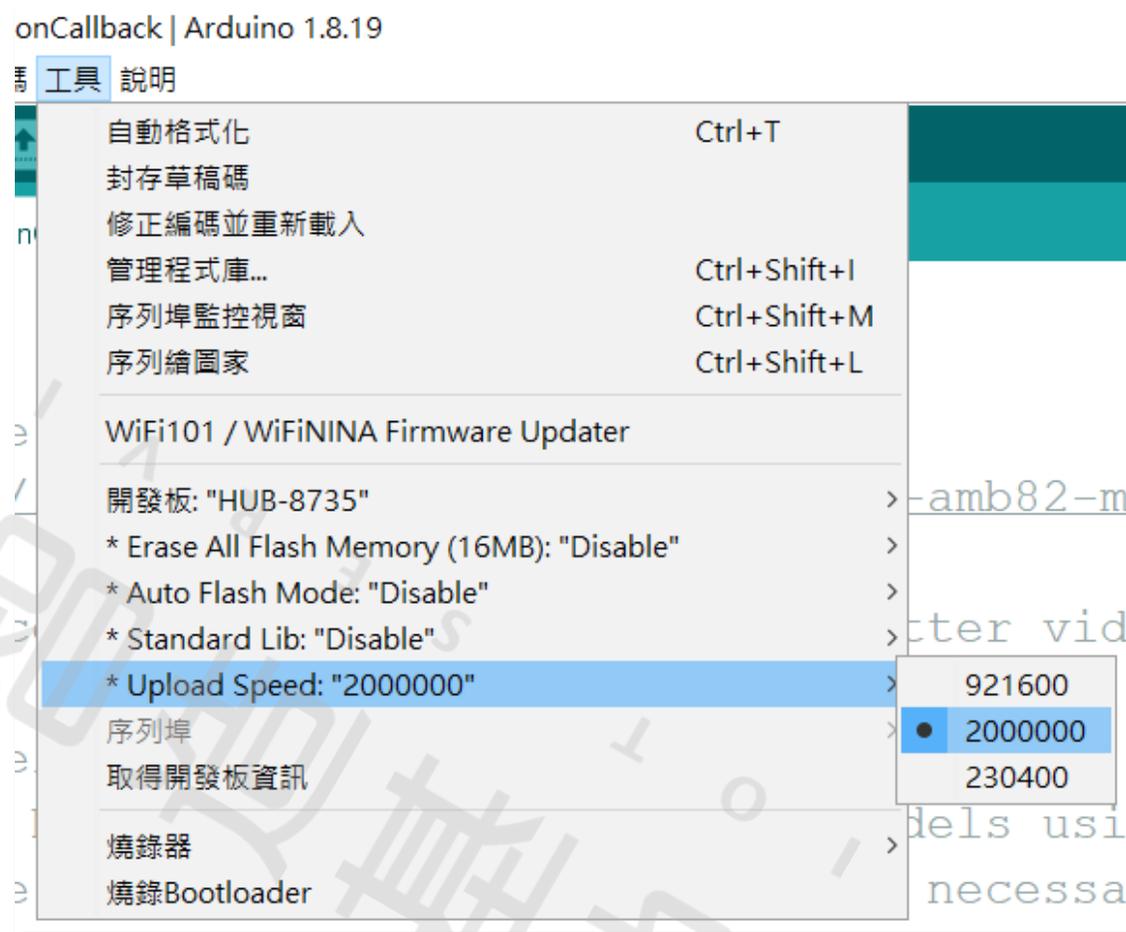


圖2. 傳輸速度選擇

以Arduino IDE操作Smart-CAM - 燒錄HUB 8735 (3)

CP210X不支援2M speed 

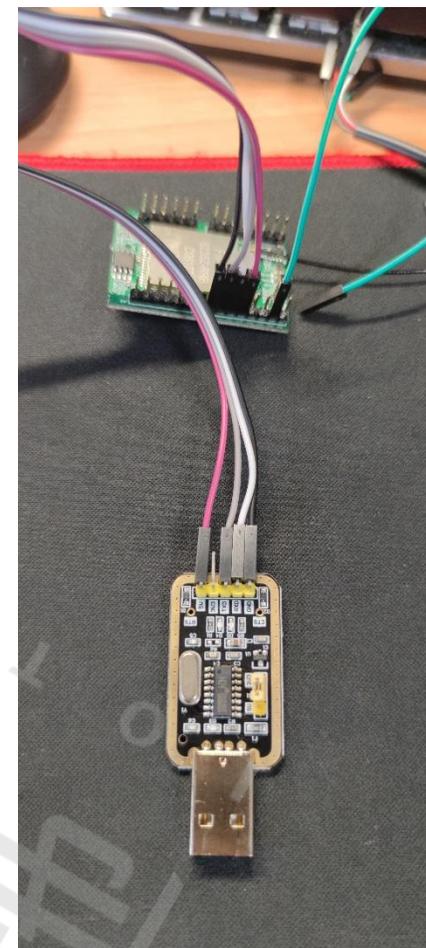
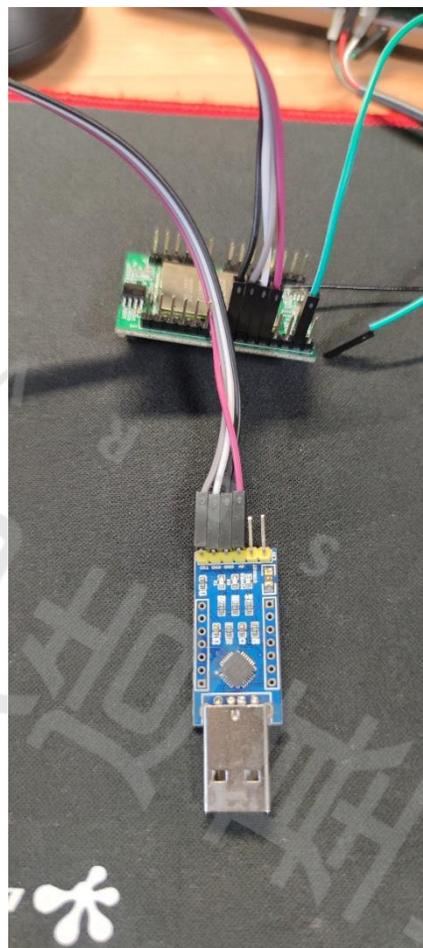
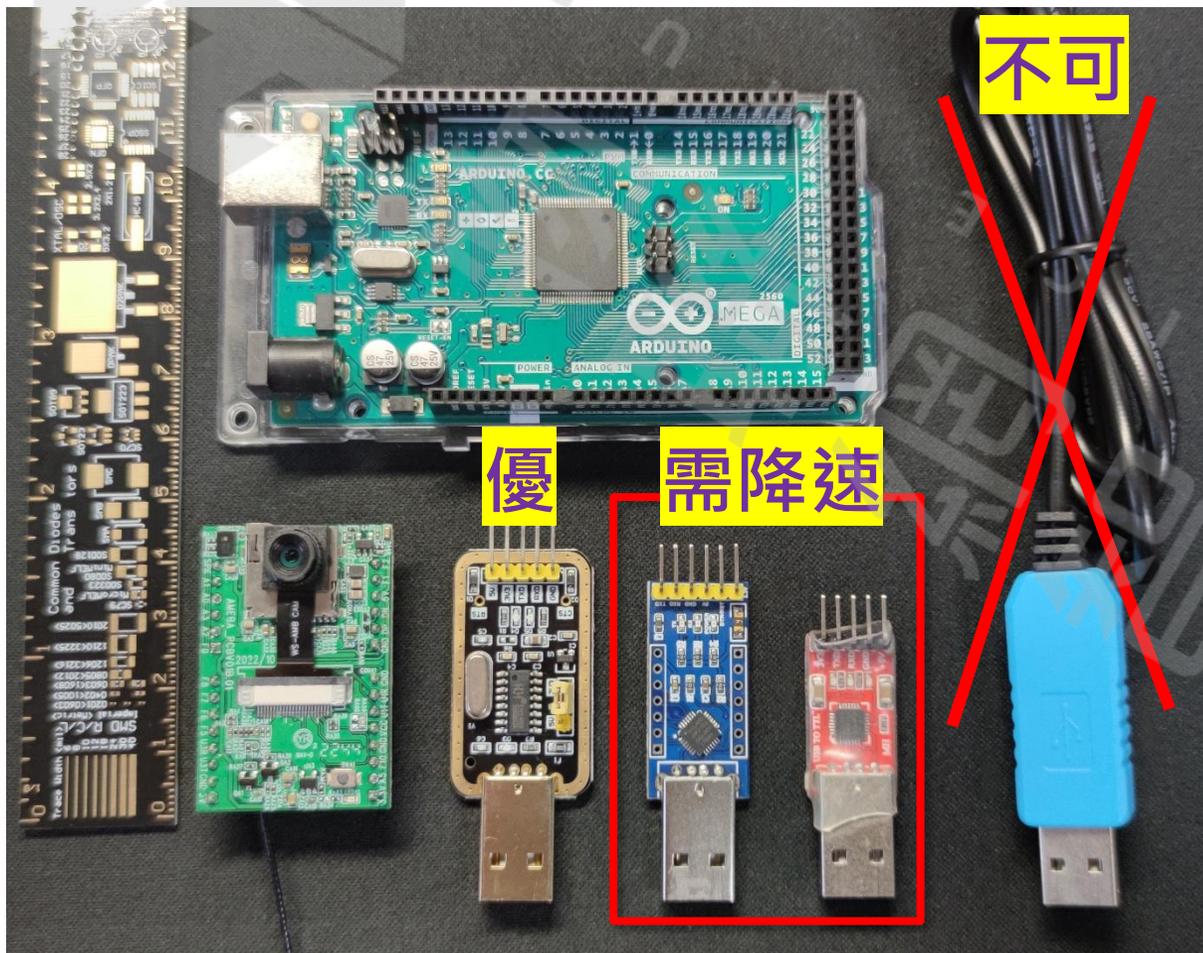


圖1~3. 各類USB to TTL

以Arduino IDE操作Smart-CAM - 燒錄HUB 8735 (4)

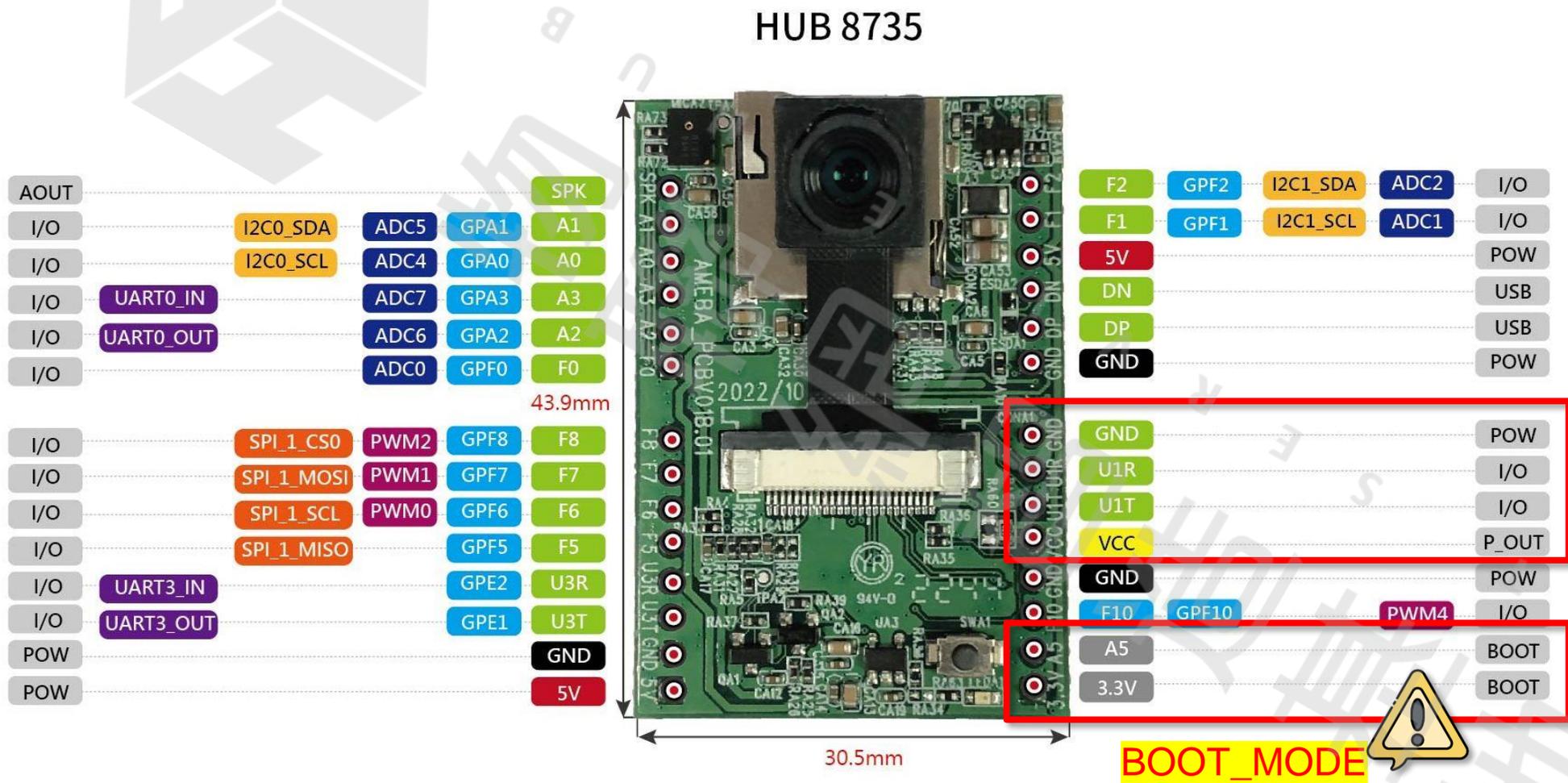


圖1. 燒入接線指示圖

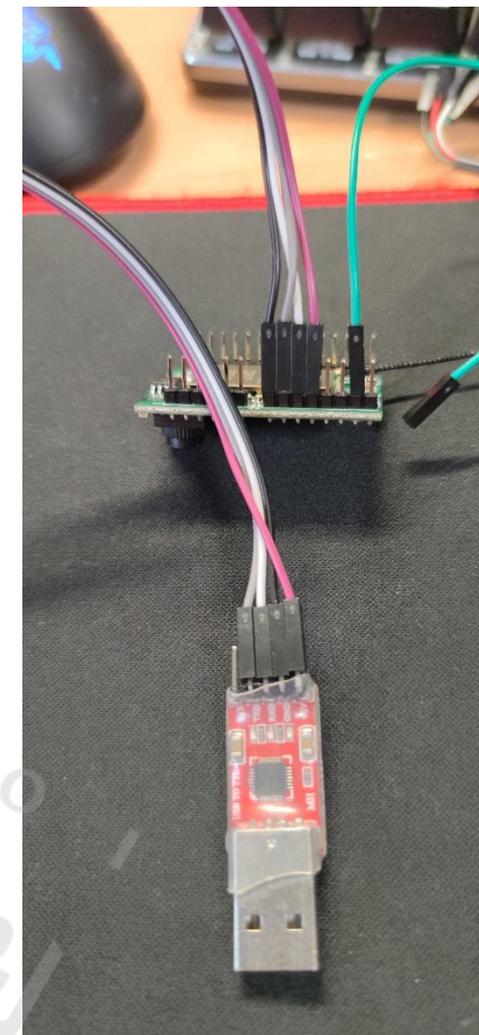


圖2. 燒入接線實體圖

以Arduino IDE操作Smart-CAM - 燒錄HUB 8735 (5)

HUB 8735

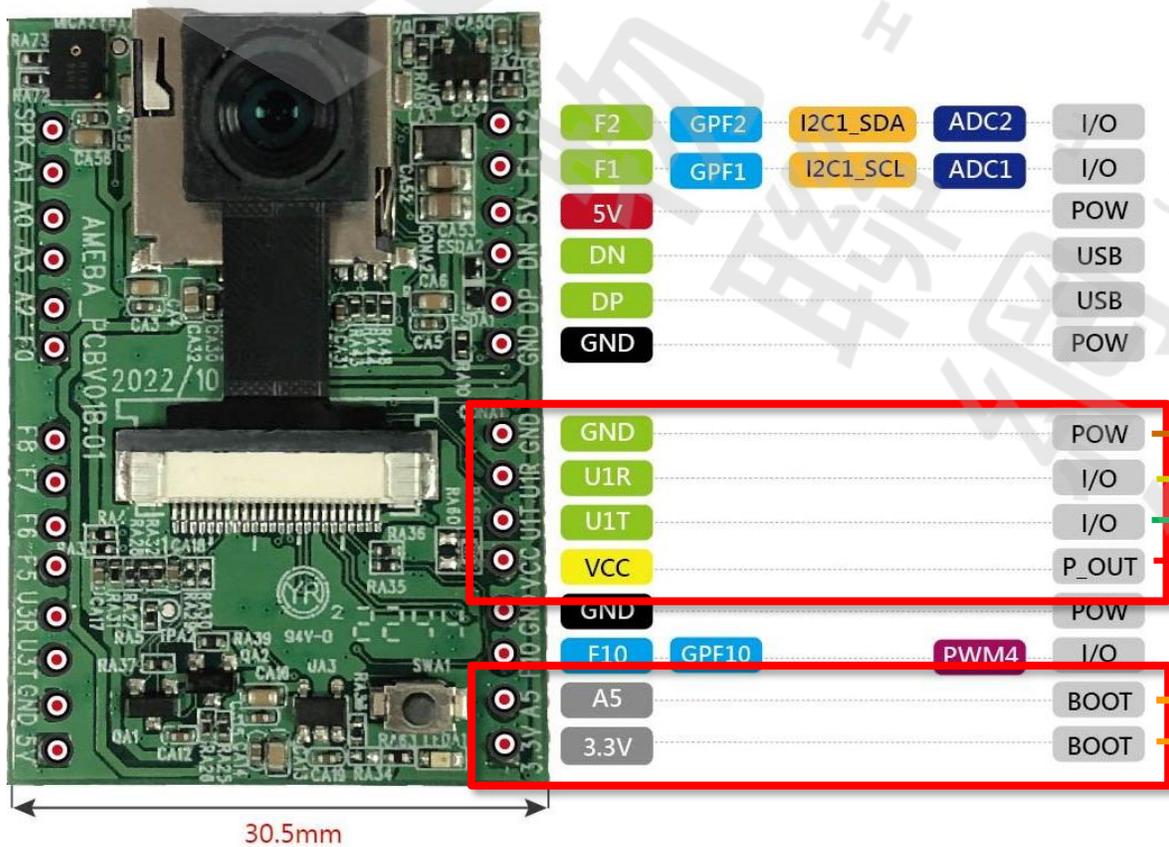


圖1. 燒入流程圖

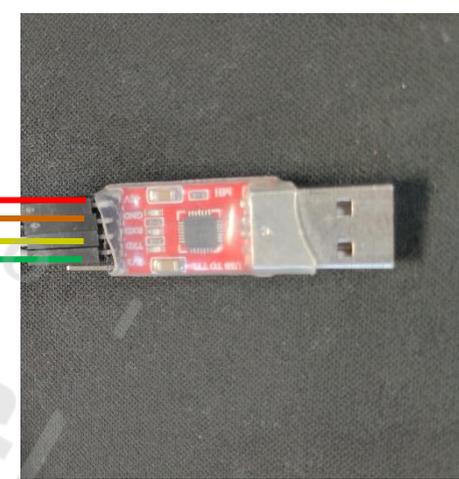


圖2. 燒入接線指示圖

、以Arduino IDE操作Smart-CAM -燒錄HUB 8735 (6)

步驟1. 下載VLC Media Player:

<https://github.com/portapps/vlc-portable/releases>

步驟2. 開啟序列埠監控視窗

步驟3. 拔掉BOOT_MODE跳線

步驟4. 按下重置鈕

步驟5. 查看ip

步驟6. 連線即時串流協定

(Real Time Streaming Protocol, RTSP)

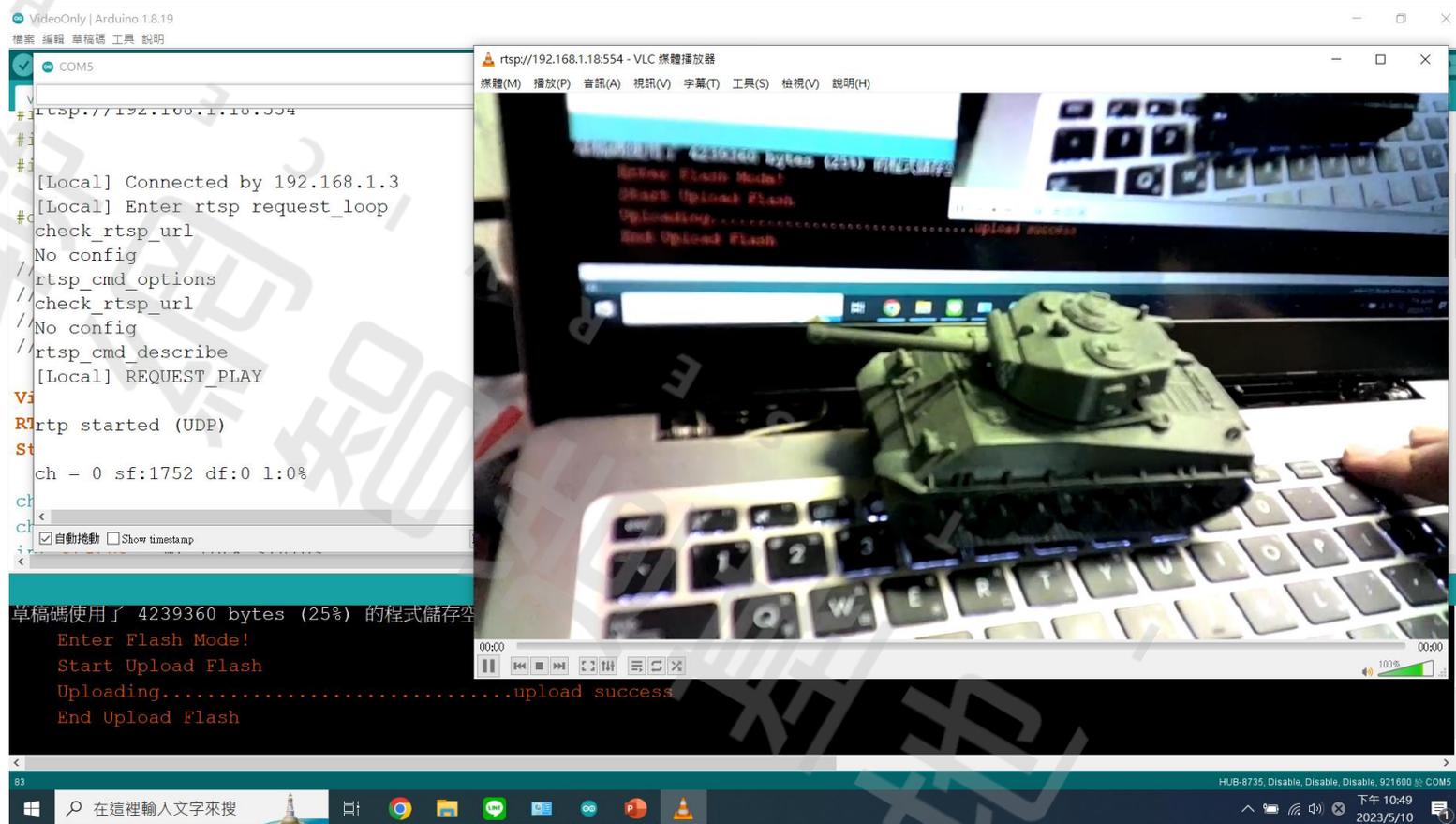
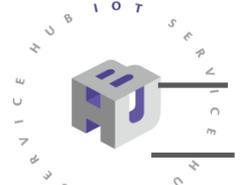


圖1. 成功畫面



以Arduino IDE操作Smart-CAM - 燒錄HUB 8735 (7)

```
VideoOnly | Arduino 1.8.19  
檔案 編輯 草稿碼 工具 說明  
COM5  
# ERROR: SET user data size [0], need more than 1  
# hal_voe_send2voe too long 51179 cmd 0x0000020  
#  
-----  
- Summary of Streaming -  
-----  
#  
Channel: 1  
Encoder type: H264  
Resolution: VIDEO_HD  
Video width: 1280  
Video height: 720  
fps: 30  
bps: 4194304  
Vi  
RTSP -  
Startsp://192.168.1.18:554  
cl  
cl  
[x] 自動捲動 [ ] Show timestamp [NL(newline)]  
草稿碼使用了 4239360 bytes (25%) 的程式儲存空間。上限  
Enter Flash Mode!  
Start Upload Flash  
Uploading.....uplo  
End Upload Flash
```

Step.2

Step.1

RTSP -
Startsp://192.168.1.18:554

VLC 媒體播放器
媒體(M) 播放(P) 音訊(A) 視訊(V) 字幕(T) 工具(S) 檢視(V) 說明(H)
開啟檔案(F)... Ctrl+O
開啟多個檔案(O)... Ctrl+Shift+O
開啟資料夾(F)... Ctrl+F
開啟光碟(D)... Ctrl+S
開啟網路串流(N)... Ctrl+N
開啟擷取裝置(C)... Ctrl+C
從剪貼簿開啟位置(L) Ctrl+V
開啟最近使用的媒體(R)
儲存播放清單至檔案(F)... Ctrl+Y
轉換(R) / 儲存... Ctrl+R
串流(S)... Ctrl+S
到達播放清單結尾時離開
離開(Q) Ctrl+Q

開啟媒體
檔案(F) 光碟(D) 網路(N) 擷取裝置(D)
網路通訊協定
請輸入網址: Step.3
rtsp://192.168.1.18:554
http://www.example.com/stream.avi
rtp://@:1234
mms://mms.examples.com/stream.asx
rtsp://server.example.org:8080/test.sdp
http://www.youtube.com/watch?v=gg64x
顯示更多選項(M) Step.4
快取 1000 毫秒
開始時間 00H:00m:00s.000
停止時間 00H:00m:00s.000
[] 同步播放另一個媒體(額外的音訊檔案, ...)
MRL rtsp://192.168.1.18:554
編輯選項 :network-caching=1000
播放(P) 取消(C)

圖1. VLC Media Player操作

以Arduino IDE操作Smart-CAM - 燒錄HUB 8735 (8)

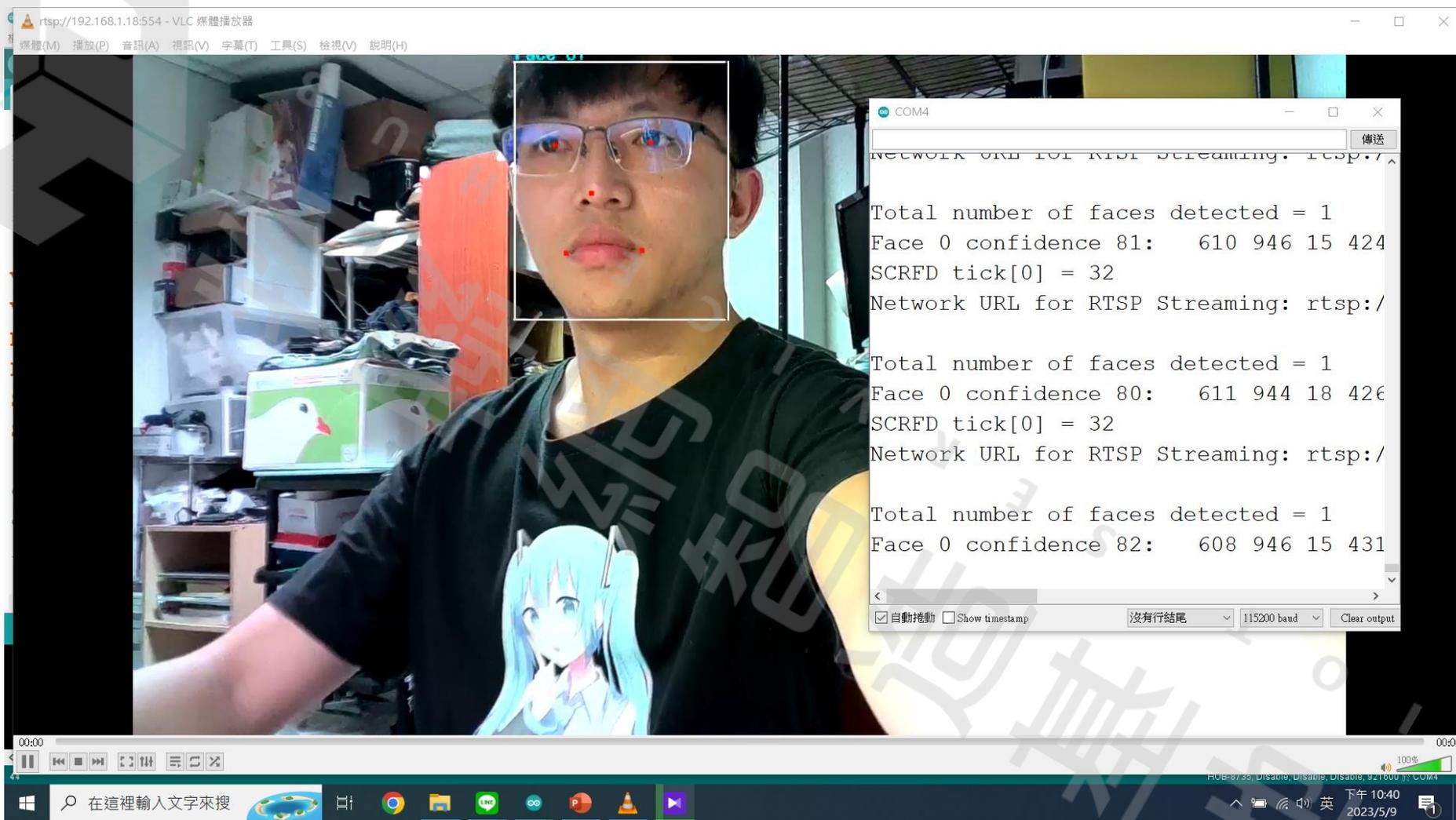


圖1. VLC Media Player操作

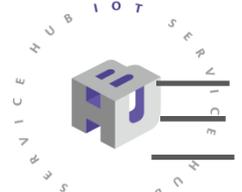
labellmg 標注工具介紹-讓AI知道這是什麼?



圖1. 數字辨識



圖2. 自動算錢



labellmg 標注工具介紹 - 蒐集照片

```
HTTPDisplayJPEG | Arduino 1.8.19
檔案 編輯 草稿碼 工具 說明

HTTPDisplayJPEG $
/* This example uses the camera to capture a JPEG image
 * and sends the image to a browser using HTTP.
 * Use any browser and connect to the board's IP address.
 */
#include <WiFi.h>
#include "VideoStream.h"

#define CHANNEL 0

// Use a pre-defined resolution, or choose to configure
VideoSetting config(VIDEO_FHD, CAM_FPS, VIDEO_JPEG, 1);
//VideoSetting config(1072, 1072, CAM_FPS, VIDEO_JPEG,

char ssid[] = "KuroNeko"; // your network SSID (name)
char pass[] = "27182818"; // your network password
int status = WL_IDLE_STATUS;
WiFiServer server(80);

<
上傳完畢
草稿碼使用了 4239360 bytes (25%) 的程式儲存空間。上限為 16111
Enter Flash Mode!
Start Upload Flash
Uploading.....upload success
End Upload Flash

15
```

圖1. 開啟HTTP顯示JPEG範例

The screenshot shows a web browser window with the address bar containing `192.168.50.226`. Below the browser is a serial monitor window displaying the following log:

```
21:44:17.950 -> [INFO] Accept connection successfully
21:44:17.950 ->
21:44:17.950 -> A client connected to this server :
21:44:17.950 -> [PORT]: 61778
21:44:17.950 -> [IP]:192.168.50.69
21:44:17.997 ->
21:44:17.997 -> new client connected
21:44:18.045 -> GET /favicon.ico HTTP/1.1
21:44:18.045 -> Host: 192.168.50.226
21:44:18.045 -> Connection: keep-alive
21:44:18.045 -> User-Agent: Mozilla/5.0 (Windows NT 10.0; Win6
21:44:18.045 -> Accept: image/avif,image/webp,image/apng,image
21:44:18.092 -> Referer: http://192.168.50.226/
21:44:18.092 -> Accept-Encoding: gzip, deflate
21:44:18.092 -> Accept-Language: zh-TW,zh;q=0.9,en-US;q=0.8,en
21:44:18.092 ->
21:44:18.139 -> client disconnected
21:44:18.754 ->
21:44:18.754 -> [INFO] Accept connection successfully
21:44:18.754 ->
21:44:18.754 -> A client connected to this server :
21:44:18.754 -> [PORT]: 61779
21:44:18.754 -> [IP]:192.168.50.69
21:44:18.754 ->
21:44:18.754 -> new client connected
```

圖2. 開啟網址並存圖

labellmg 標注工具介紹 - 操作軟體



<https://github.com/heartexlabs/labellmg/releases>

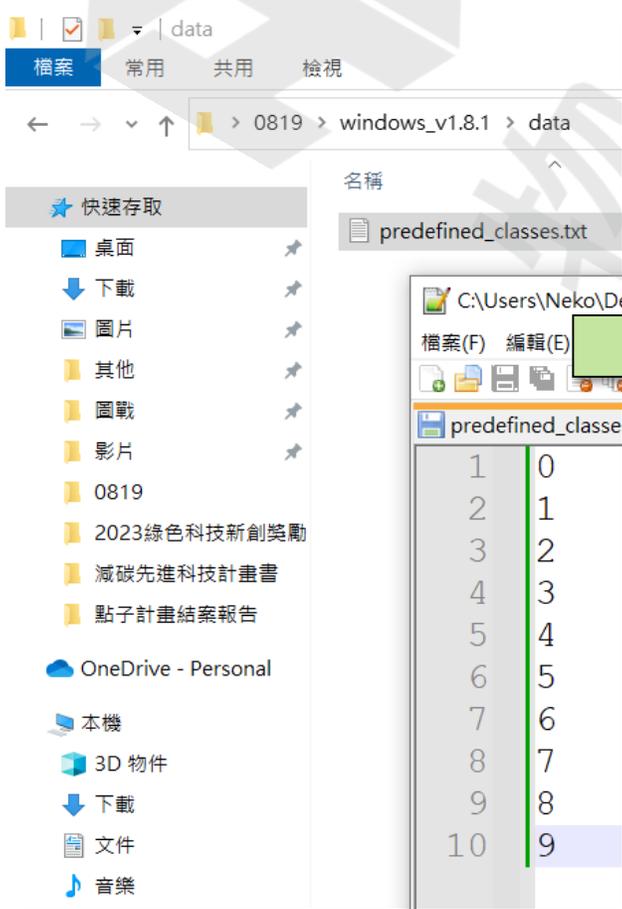


圖2. 本次的設定



圖3. labellmg

四、使用Google Colab製作AI模型 -上傳數據

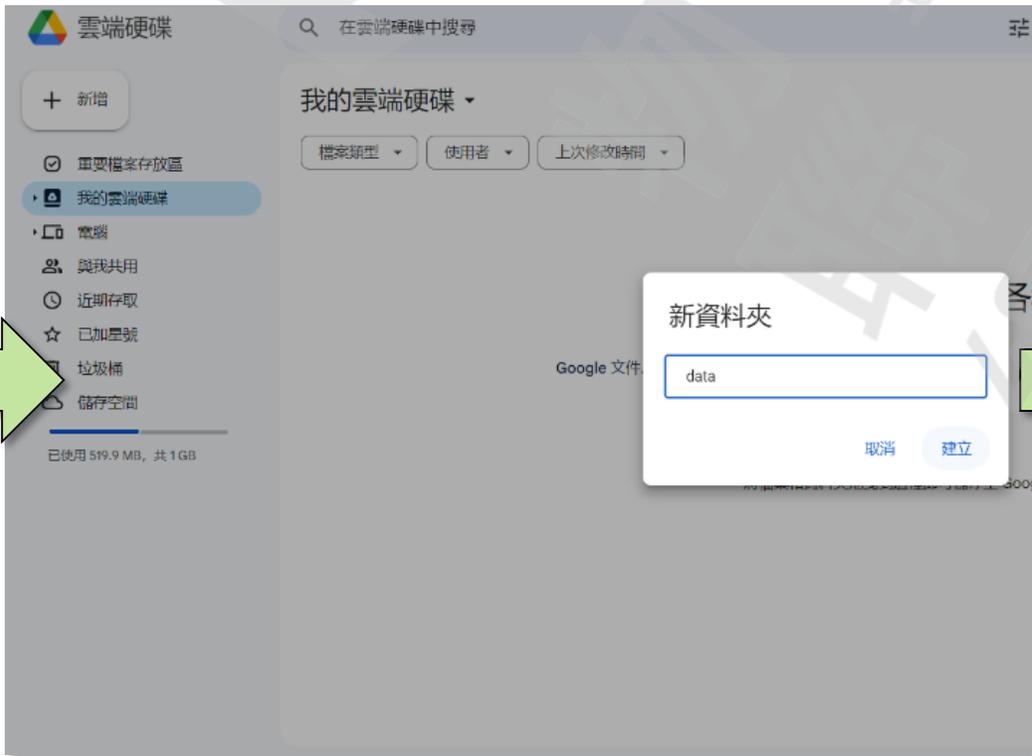


圖1.按右鍵新增 'data' 資料夾

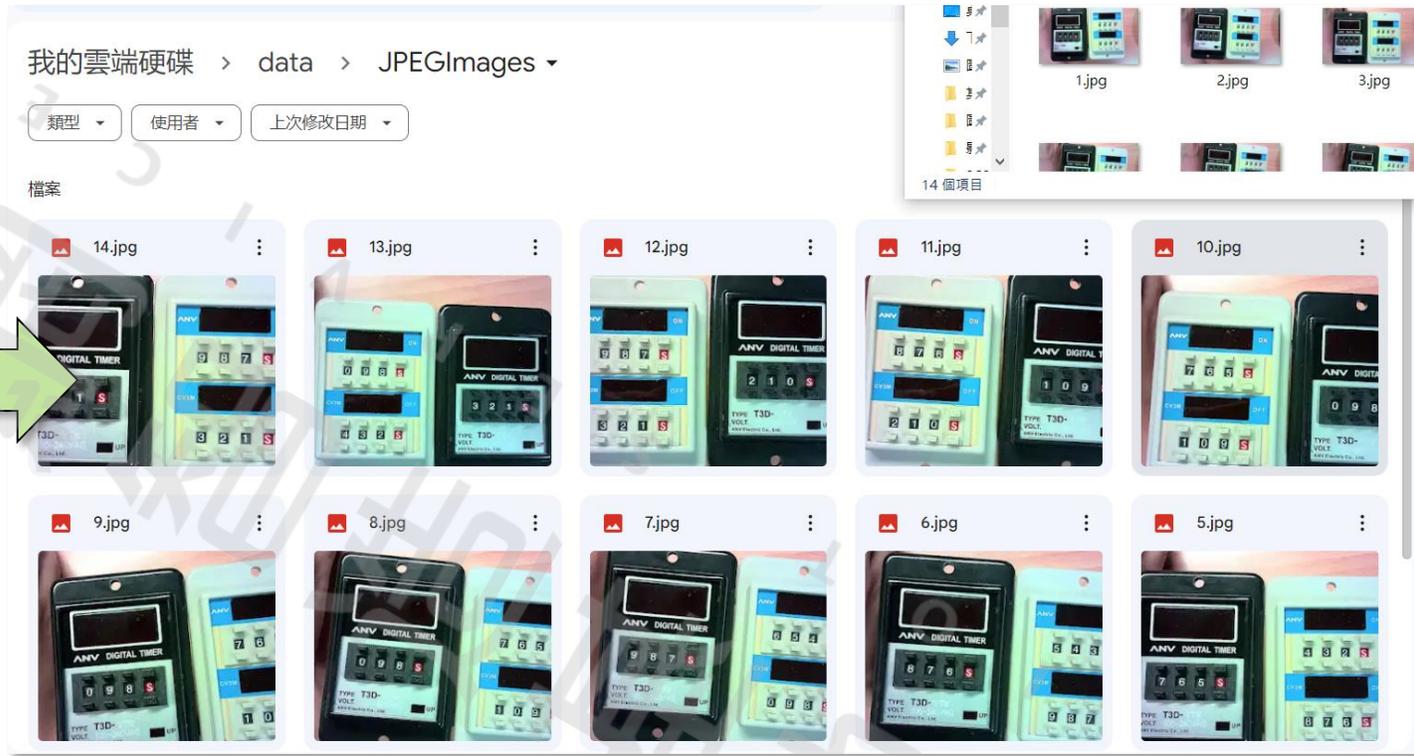


圖2. 'data'資料夾下新增 'JPEGImages'資料夾及上傳資料

四、使用Google Colab製作AI模型- 前置準備

上傳2個檔案

- Data_Augmentation_CLODSA.ipynb
- Train_YOLO_Object_Detection_With_Colab.ipynb

<https://colab.research.google.com/>

Step.1



圖1.登入Google Colab

Step.2



圖2.點選'上傳'

Step.3

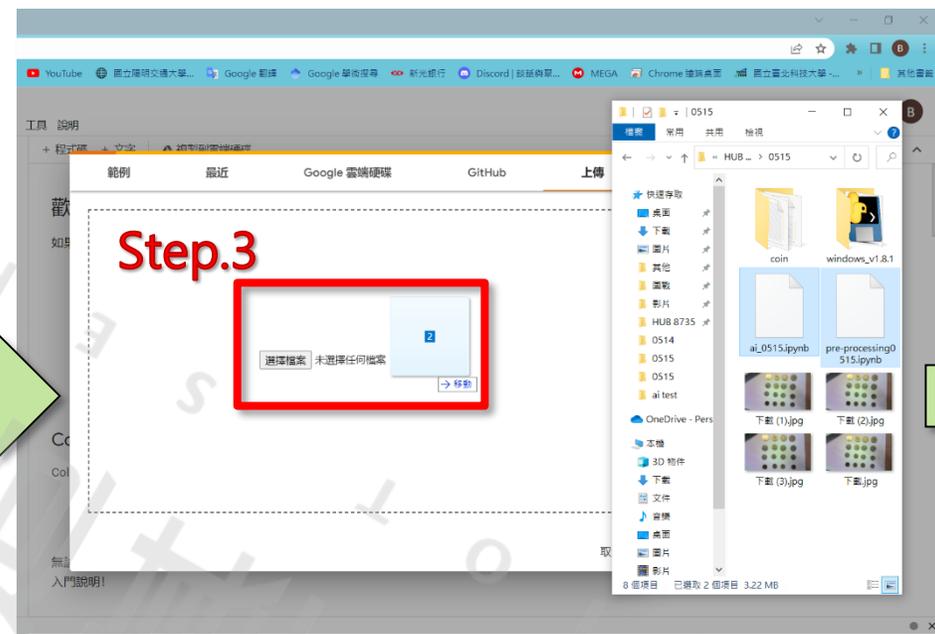


圖3.上傳準備好的範例

四、使用Google Colab製作AI模型- 資料前處理1

執行Data_Augmentation_CLODSA.ipynb



圖1. 全部執行



圖2. 點選連接至google雲端硬碟



圖2. 允許連接

四、使用Google Colab製作AI模型 – 資料增強

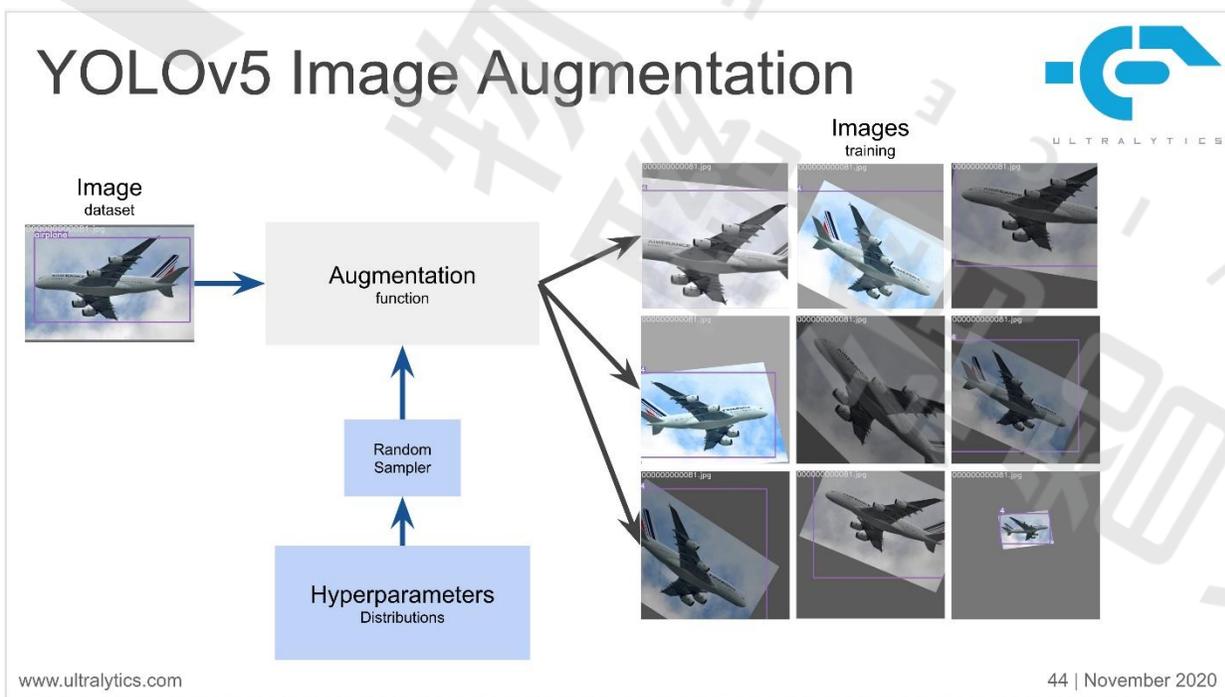


圖1.YOLO資料增強

圖片來源:<https://github.com/ultralytics>



圖2. 影像翻轉、拉伸、噪點處理

四、使用Google Colab製作AI模型-資料前處理2

```

前處理0514.ipynb
檔案 編輯 檢視畫面 插入 執行階段 工具 說明
+ 程式碼 + 文字
40 秒
✓ [1] %cd ..
from google.colab import drive
drive.mount('/content/gdrive')

Mounted at /content/gdrive

[2] %cd /content/gdrive/MyDrive/data
%ls

/content/gdrive/MyDrive/data
JPEGImages/ labels/

import os
import random

trainval_percent = 0.1
train_percent = 0.9
txtfilepath = 'labels'
total_txt = os.listdir(txtfilepath)

num = len(total_txt)
list = range(num)
tv = int(num * trainval_percent)
tr = int(tv * train_percent)
trainval = random.sample(list, tv)
train = random.sample(trainval, tr)

#xvalval = os.path.join('xvalval.txt', 'w')
執行中 (已持續 0 秒) <col>

```

圖1. 確認執行結果

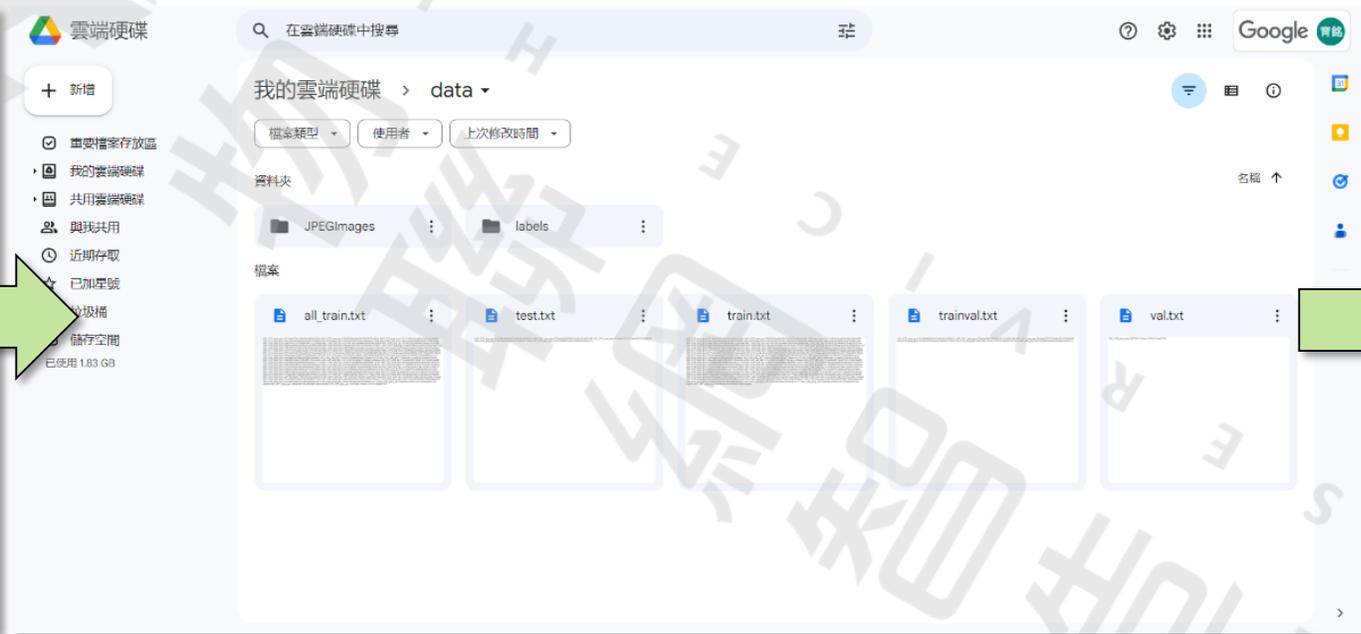
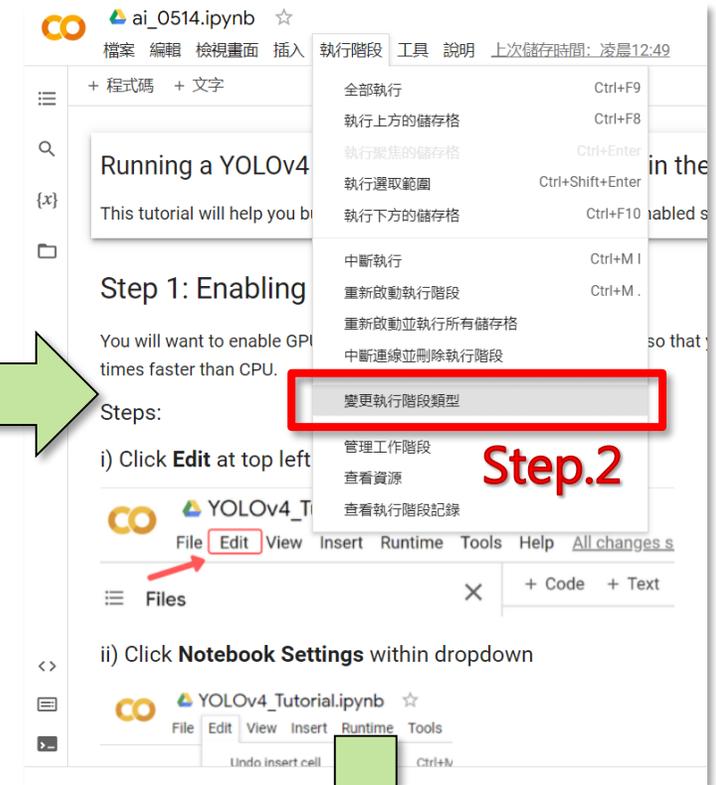
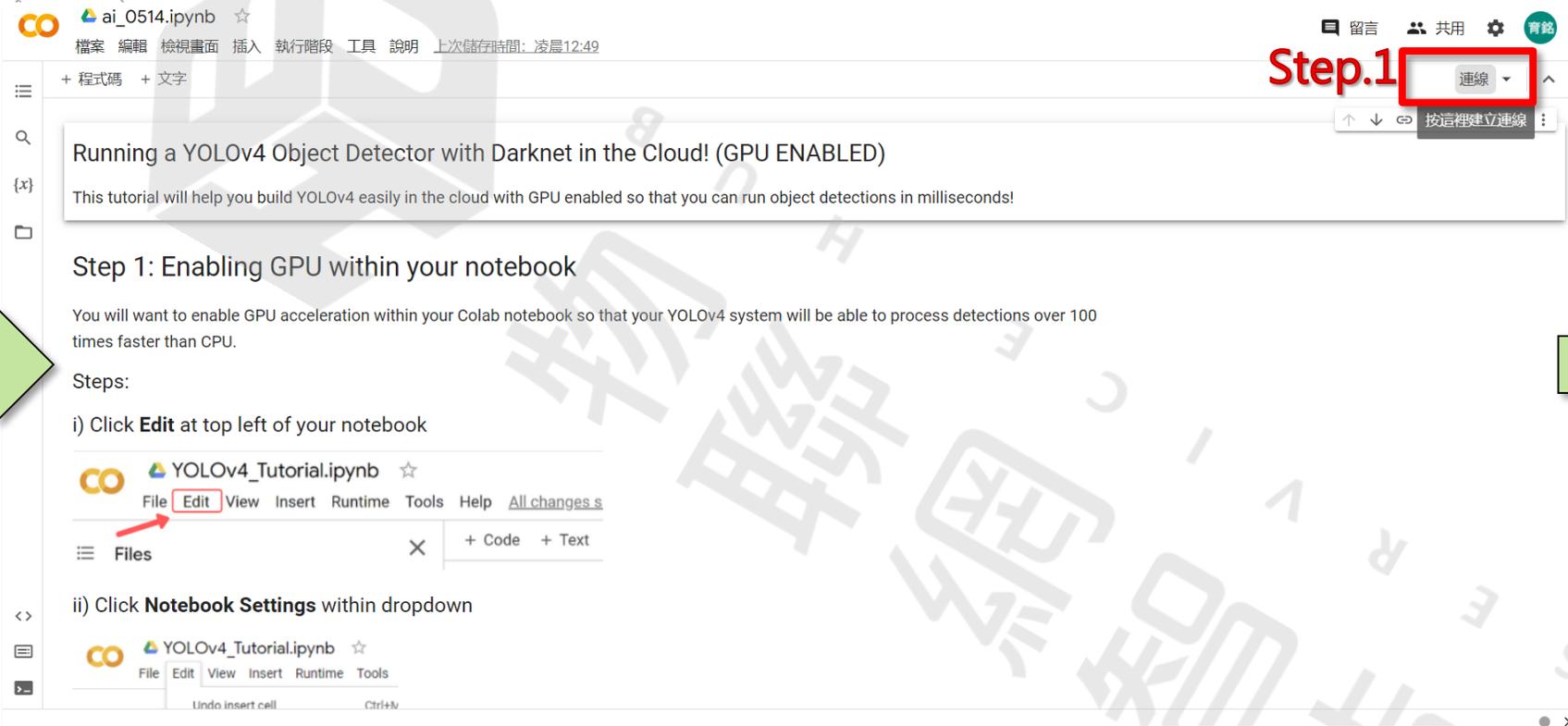


圖2. 執行後應出現5個.txt



圖3. 點選中斷連線

四、使用Google Colab製作AI模型-設定GPU運算模式



資料:

<https://github.com/AlexeyAB/darknet>

參考:

https://colab.research.google.com/drive/12QusaaRj_IUwCGDvQNflCpa7kA7_a2dE

圖1~3. 使用Colab GPU方式運算

四、使用Google Colab製作AI模型- 換自己的資料集1



圖1. 執行連線Google Drive

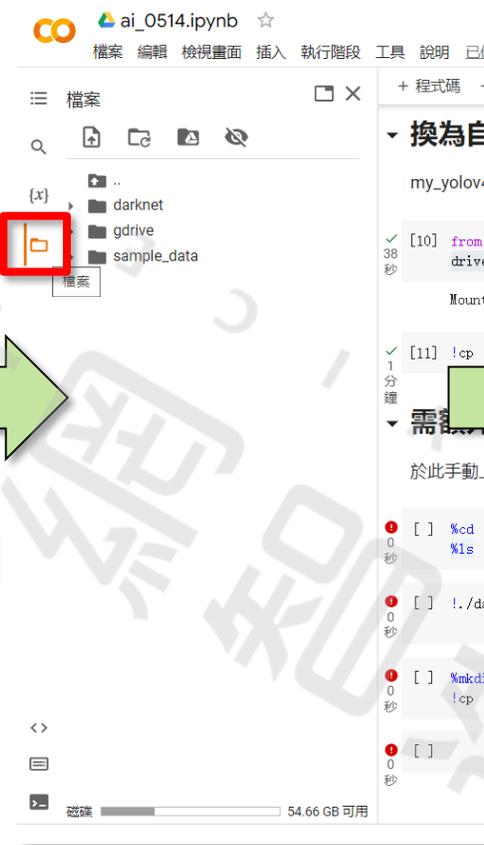


圖2. 開啟資料夾



圖3. 拖曳檔案上傳



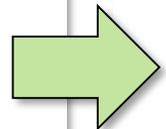
四、使用Google Colab製作AI模型- 換自己的資料集2

```
[net]
# Testing
batch=1
subdivisions=1
# Training
#batch=64
#subdivisions=1
width=416
height=416
channels=3
momentum=0.9
decay=0.0005
angle=0
saturation = 1.5
exposure = 1.5
hue=.1

learning_rate=0.00261
burn_in=1000

max_batches = 2000200
policy=steps
steps=1600000,1800000
scales=.1,.1

#weights_reject_freq=1001
#ema_alpha=0.9998
#early_stopping_patience=1000
```



測試模型

手動上傳 my_yolov4-tiny_test.cfg 和測試照片test.jpg

```
# run darknet detection on test images
!. /darknet_detector test my_ai.data my_yolov4-tiny...
```

YOLO_Object_Detection_With_Colab.ipynb

11111
lab Notebooks
ta
JPEGImages
1.jpg
10.jpg
11.jpg
12.jpg
13.jpg
14.jpg
2.jpg

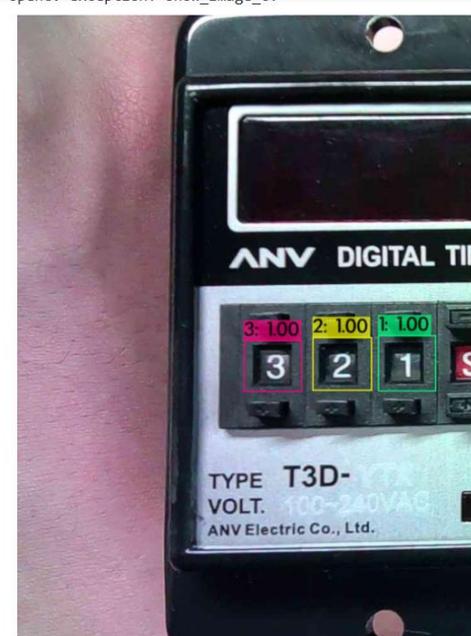
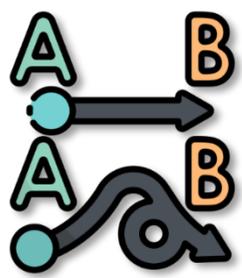


圖1. 新增及修改.cfg

圖2. 上傳測試模型設定與測試照片

圖3. 成功辨識



成功編譯

四、使用Google Colab製作AI模型- 換自己的資料集2

The screenshot shows two windows in Notepad++:

- Left Window (my_ai_names):** Contains a list of numbers from 0 to 9. A red box highlights the entire list.
- Right Window (my_yolov4-tiny.cfg):** Contains configuration parameters for a YOLOv4 model.
 - Line 221: `filters=45` is highlighted with a red box and labeled **Step.2**. Below it, a yellow box contains the formula: $Filters = 3 * (5 + len(classes))$.
 - Line 229: `classes=10` is highlighted with a red box and labeled **Step.3**.
 - Line 226: `[yolo]` is highlighted with a red box and labeled **Step.1**.

A search dialog box is open over the right window, showing the search term "yolo" and the "Find Next" button.

圖1. 修改.cfg 跟 .names

四、使用Google Colab製作AI模型- 換自己的資料集3

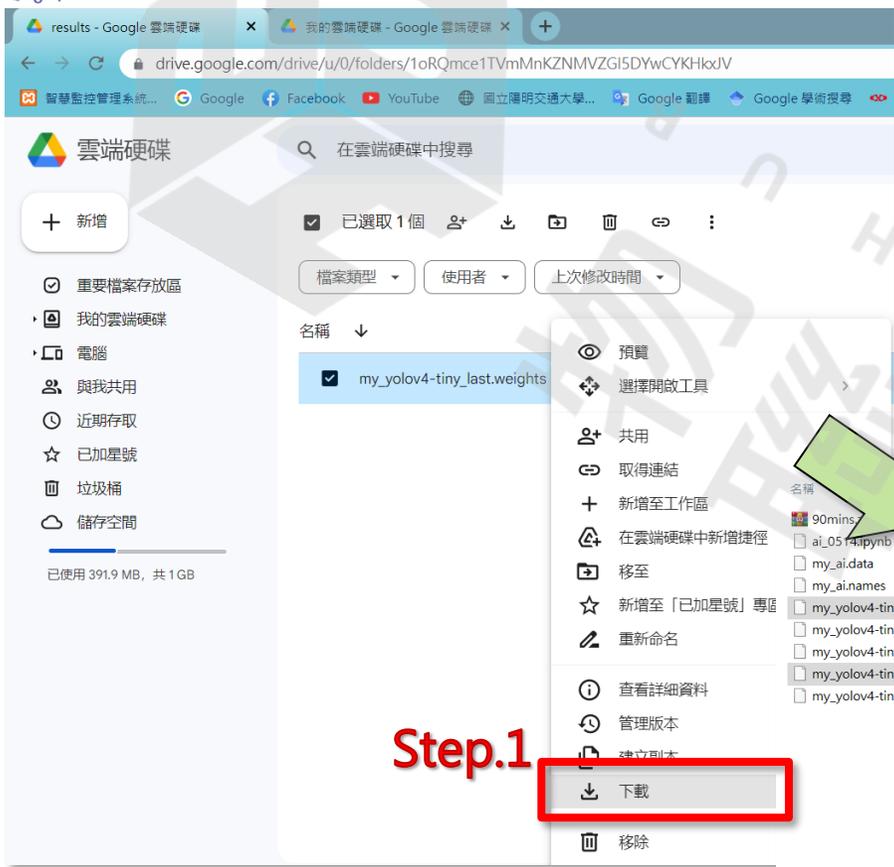


圖1. 下載權重

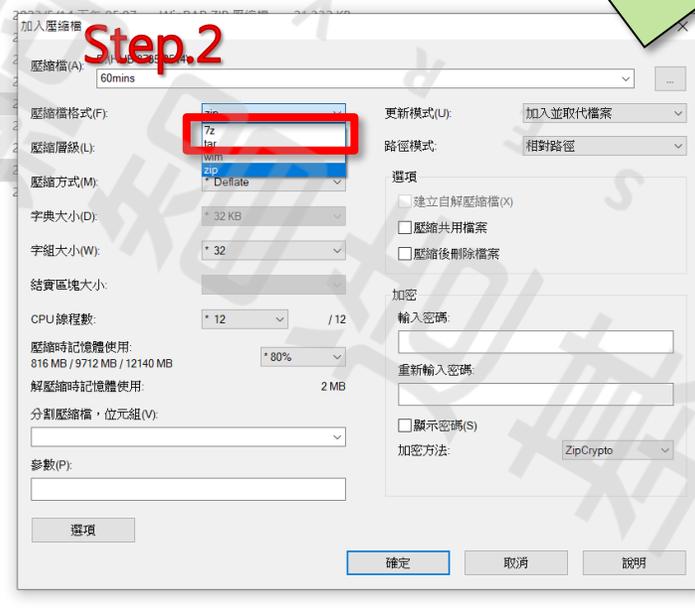


圖2. 壓縮為'.ZIP'檔案

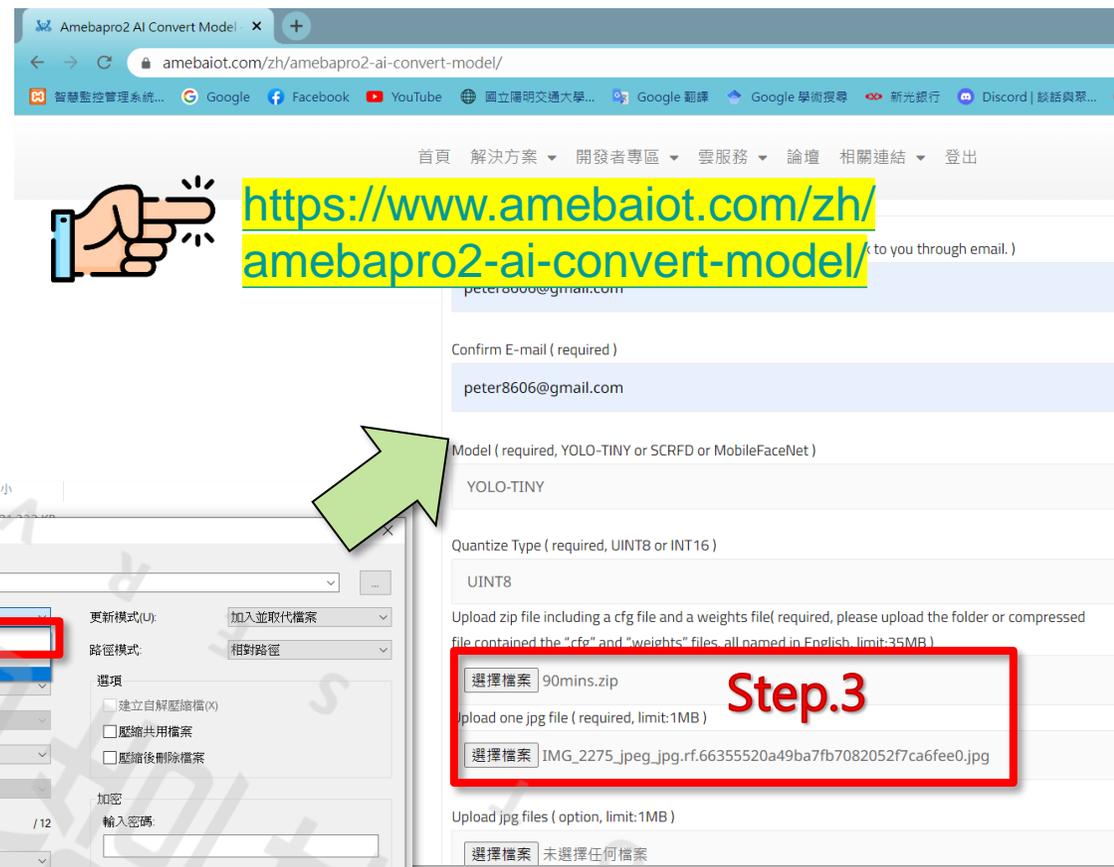


圖3. 檔案上傳轉換



四、以Arduino IDE將模型燒入晶片-燒入自己的AI模型

步驟1. 等待自動回覆至登記註冊 e-mail 信箱裡，並將 nb 檔案下載。或是如下網址:

https://www.amebaiot.com/wp-content/uploads/MQTT/<使用者ID>/network_binary.nb

步驟2. 開啟資料夾至:

```
C:\Users\<<使用者名稱>  
>\AppData\Local\Arduino15\packages\realtek\hardware\AmebaPro2\4.0.3\variants\common_nn_models
```

步驟3. 替換 yolov4_tiny.nb 檔案。

參考路徑:

https://www.amebaiot.com/wp-content/uploads/MQTT/wildman8606/network_binary.nb

```
C:\Users\Neko\AppData\Local\Arduino15\packages\ideasHatch\hardware\AmebaPro2\4.0.3-  
build20230510\variants\common_nn_models
```

五、以Arduino IDE將模型燒入晶片-燒入自己的AI模型

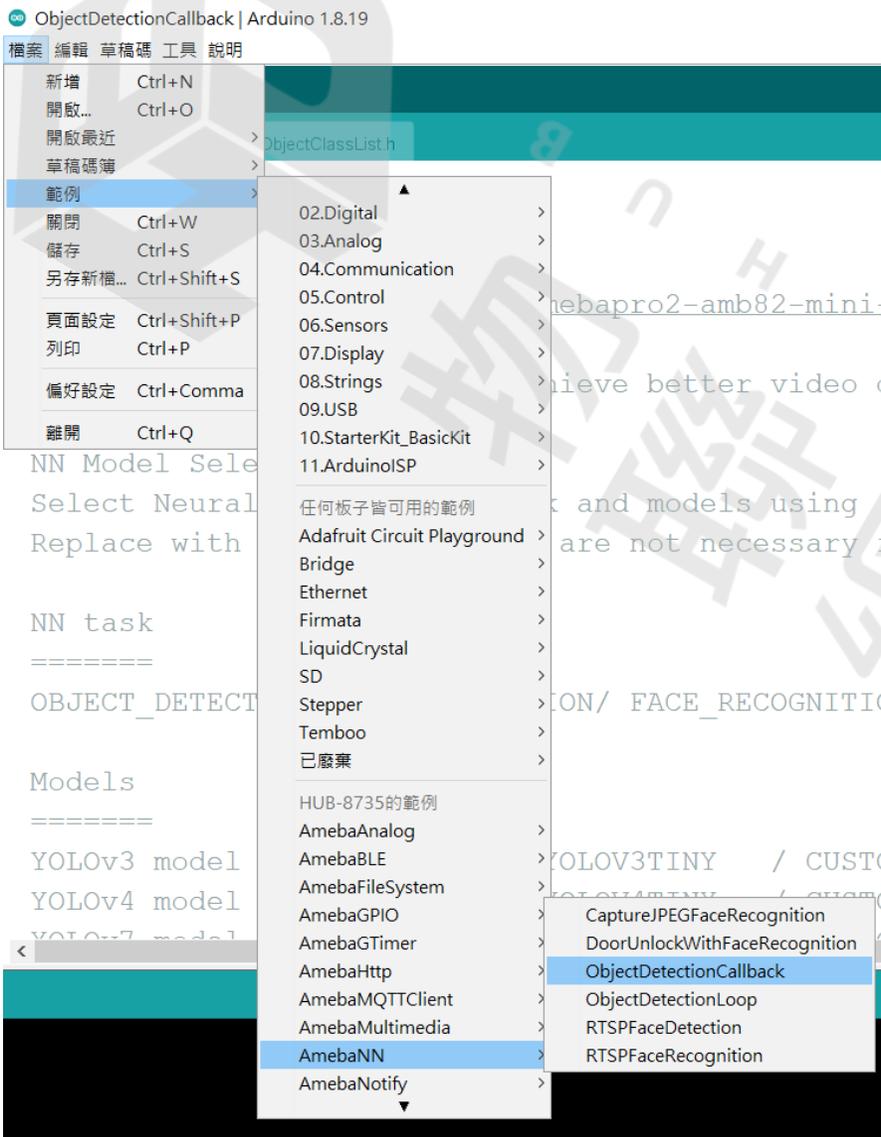


圖1. 開啟範例

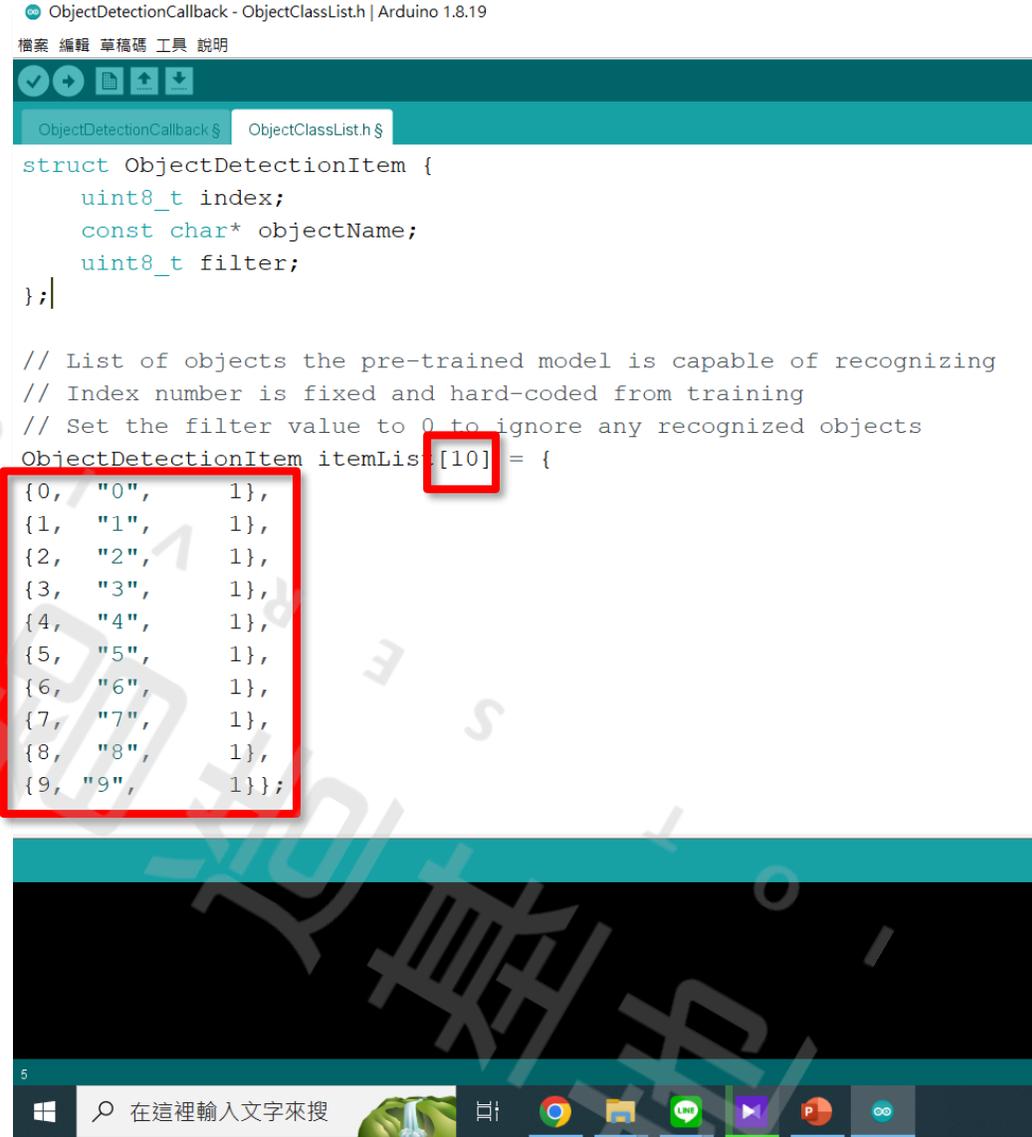
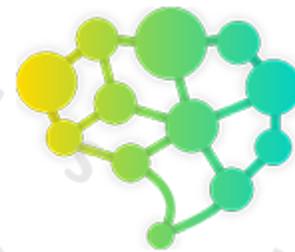
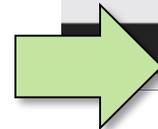
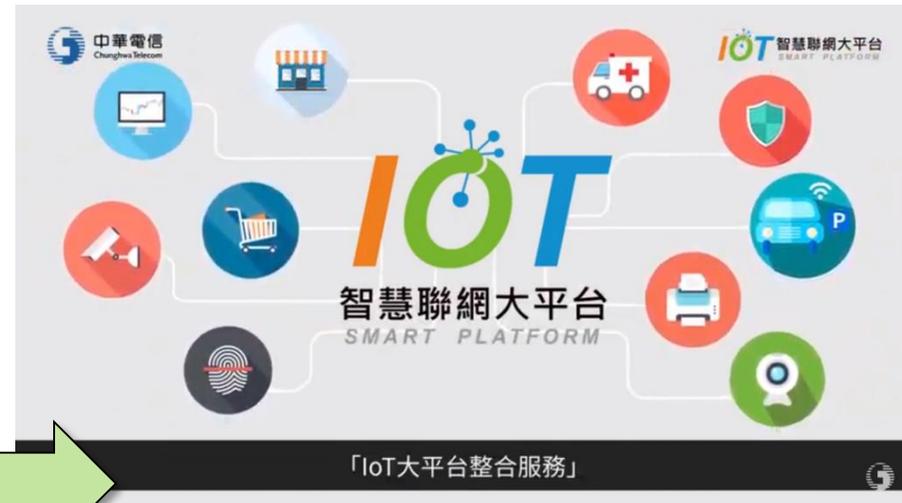
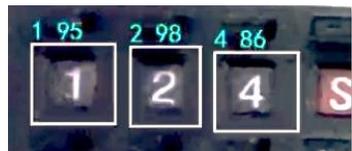


圖2. 修改參數

六、補充 - 上傳雲端資料



IDEASChain



RTSP
Real-Time Streaming Protocol



圖1. HUB8735工業物聯網

六、補充 - 其他案例手把手教學



圖1. 開啟範例

☰ README.md

HUB 8735 國產IC應用實作開發系列活動

大家好，簡單好操作的HUB 8735 (AMB82 MINI、AmebaPro2)的手把手教學自製模型於19號於資策會有開課教學。我這邊有重新錄製當天教學內容，有需要可配溫開水服用，之後還會推出多個案例~

新一代AIoT Camera方案- HUB 8735，可直接運行多款AI模型，模組精簡容易自行開發，輕鬆替換一般IoT Camera。由國內瑞昱大廠與資策會極力推廣，比ESP32-CAM具有AI功能的升級，增加內置NPU AI 運算引擎。

HUB 8735實務Training影片教學: <https://www.youtube.com/watch?v=YZafxlf89aA>

時間	議程
00:00:00 - 00:09:09	前言
00:09:12 - 00:37:30	使用Google Colab製作AI模型(水下生物辨識)
00:37:30 - 01:23:27	以Arduino IDE將模型燒入晶片(水下生物辨識)
01:23:27 - 01:37:54	labellmg標注工具介紹(硬幣辨識)
01:37:54 - 01:52:20	使用Google Colab製作AI模型(硬幣辨識)
01:52:20 - 02:20:24	補充資料
02:20:24 - 00:00:00	以Arduino IDE將模型燒入晶片(硬幣辨識)

HUB 8735 Smart AI CAM是具備多功能影像處理的高度集成模組，內置NPU AI 運算引擎，加速處理AI模型以及802.11 a/b/g/n 雙頻Wi-Fi與BLE低耗電藍牙傳輸，可廣泛應用於各種結合影像識別或 AI運算之物聯網場域，適用於智能家居，工業物聯網，智慧零售，健康照護或是車用電子等場景；多款Pre-trained AI models已最佳化在模組直接運行，可做為AI教學之體驗工具；同時尺寸僅30.5x43.9mm，亦為可直接整合在產品設計中做為快速導入Edge AI應用的快製套

圖2. 教學內容

<https://github.com/wildman8606/HUB-8735-AMB82-Mini-AmebaPro2-tutorial>



Thank you

