



物聯網智造基地

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

國產IC開發套件-HUB8735

示範案例-昆蟲辨識與分類

指導單位： 經濟部工業局
INDUSTRIAL DEVELOPMENT BUREAU
MINISTRY OF ECONOMIC AFFAIRS
經濟部工業局

執行單位：財團法人資訊工業策進會

作者：宅老大數位 陳淨騰 Kevin

 資訊工業策進會 Institute for Information Industry



- 方案介紹

- AI Model 訓練過程

- 程式設計

- 成果



Ideas Hatch



方案介紹

Ideas Hatch



方案介紹

● 環境與生態監測

- ◆ 昆蟲是生態系統中重要的指標物種，能夠反映出環境的健康狀況。通過對昆蟲物種的辨識與分類，可以了解環境變遷、污染程度及生態多樣性情況。例如，當某些昆蟲的數量驟減，可能代表環境遭到破壞，反之亦然。

● 農業管理

- ◆ 昆蟲在農業中扮演著兩個關鍵角色->作為害蟲和益蟲。
- ◆ 害蟲可以幫助農民及時採取防治措施，減少農作物損失。益蟲則有助於利用它們來增進農業產量，如蜜蜂的授粉作用。影像辨識技術提供了一個高效且準確的自動化解決方案。

● 疾病控制與公共衛生

- ◆ 些昆蟲，如蚊子、跳蚤等，會攜帶傳染病病毒或細菌，對人類健康構成威脅。有助於疾病控制機構採取早期預防措施，減少疾病傳播風險。影像辨識技術能夠協助公共衛生機構識別並追蹤這些昆蟲的活動範圍。

方案介紹

現今的台灣，有很多的對環境保育的熱情義工。默默地進行將台灣特有的昆蟲收集成為昆蟲圖鑑，做為後面的教學、環保、農業等很好的基礎。在這 AI 的時代應該也要將這領域更加地運用。



影像來自
站長嘎嘎(林義祥)編輯

資料來源：<http://gaga.biodiv.tw/new23/cp02.htm>

方案介紹

本專案則參考使用「嘎嘎昆蟲網」做為影像辨識的基礎。做為示範，選擇五種科屬，分別是牙蟲 ([Hydrophilidae](#))、星天牛 ([Cerambycidae](#))、椿象 ([Pentatomidae](#))、瓢蟲 ([Coccinellidae](#))、鍬形蟲 ([Lucanidae](#))。

以影像辨識的基礎，需要大量的圖像資料庫。若只靠圖鑑則是遠遠不夠，則透過本次案例為拋磚引玉，希望給有志之士做為參考。



HUB-8735-Ultra 介紹

❖ 開發背景

- 由資策會物聯網智造基地設計推出，以台灣IC晶片模組為核心的開發公板

❖ 主要特點

- 內置NPU AI運算引擎:加速AI模型處理
- 多種連接性:支援2.4G/5G雙頻Wi-Fi與BLE低耗電藍牙傳輸

❖ 應用領域

- 教育科學、智能住宅、工業物聯網、智慧零售、健康照護

❖ 支援AI模型

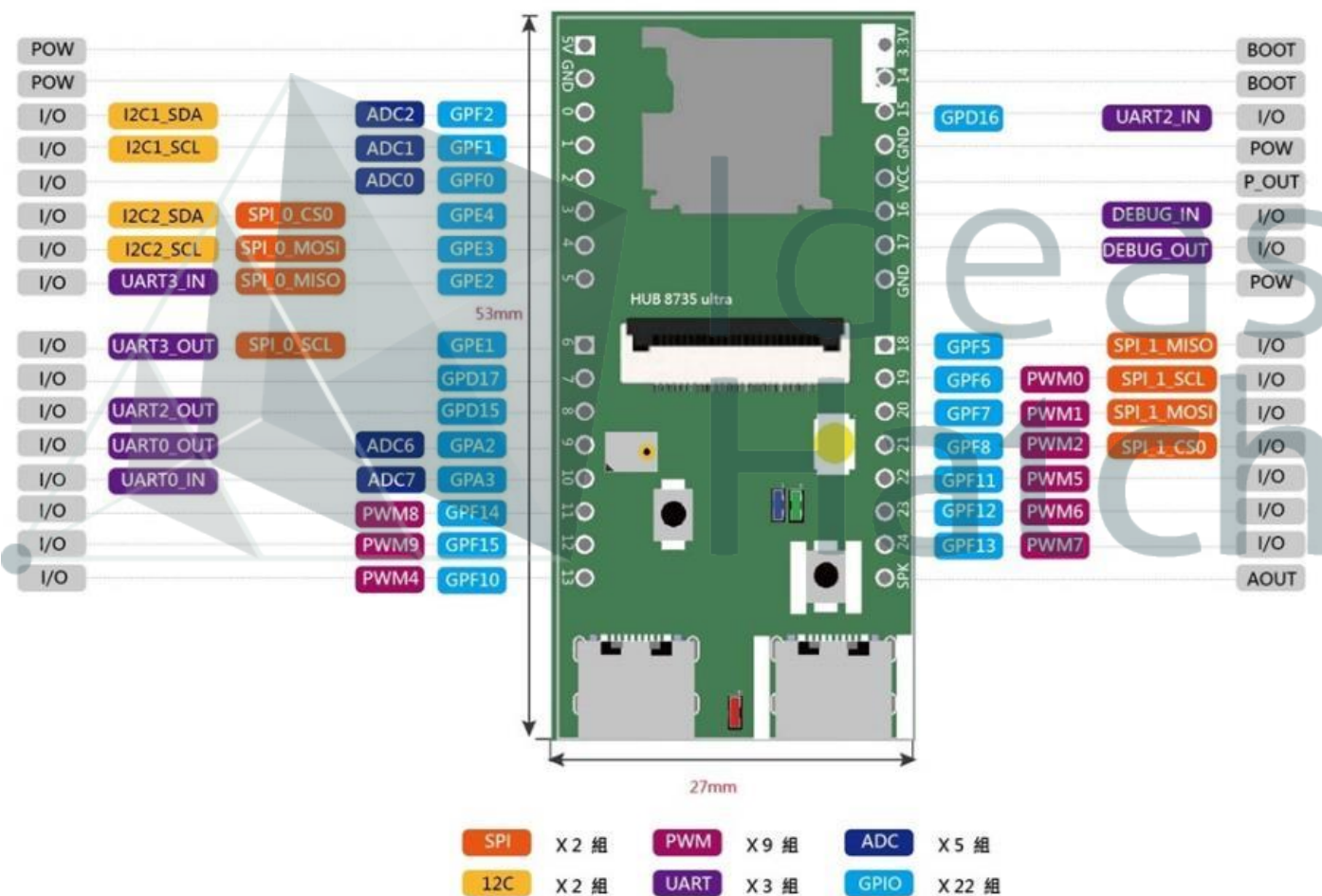
- 預先訓練好的AI模型已優化，可直接運行;支持自訓練AI模型，適用於AI教學和產品設計;可作為快速導入Edge AI應用的快製套件

HUB 8735 ultra

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 高感度數位 MIC
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	提供多種預先訓練的AI models供快速上手 (人臉辨識、物件辨識、聲音分類) 可建立自己訓練的AI 模型
硬體擴充介面	提供多組GPIO、UART、SPI、I2C、PWM、ADC擴充接腳
USB-C 介面	USB 燒錄、debug USB 影像輸出
LED	補光 LED
I/O 擴充板	依照開發者需求擴充功能，如 IMU 感測器、喇叭語音輸出、溫度、震動、濕度等功能

Ideas Hatch

HUB 8735 ultra





AI Model 訓練過程

Ideas
Hatch



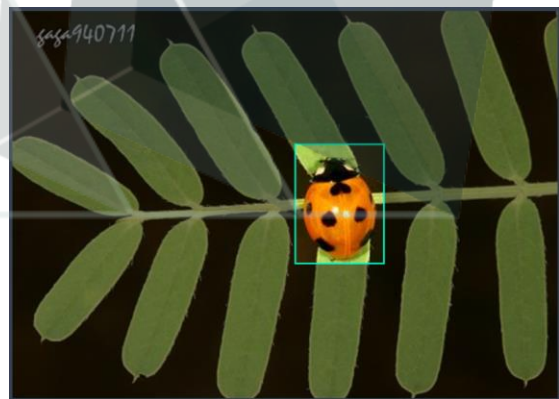
AI Model 訓練過程

因從圖鑑取得的圖像資料有限，在影像訓練資料來說是非常不夠的。所以我們利用 Roboflow AI 影像訓練(roboflow.com) 的模型增強技術，產生更多的圖像。

以下是部分的圖像範例



鍬形蟲(Lucanidae)



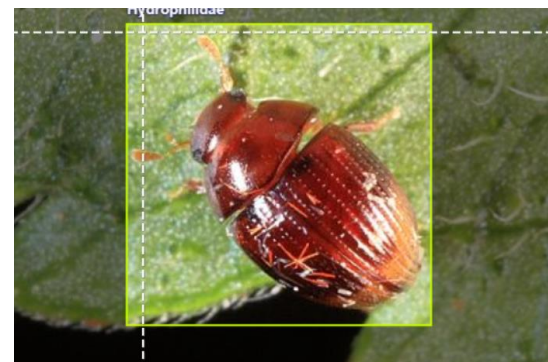
瓢蟲(Coccinellidae)



椿象(Pentatomidae)



星天牛(Cerambycidae)



椿象_Pentatomidae

AI Model 訓練過程

透過 Roboflow 的影像特徵標註工具，這個工具可以提供 HUB8738 Ultra 所需要的 YOLO 的文件產出





AI Model 訓練過程

導入 Darknet 前準備工作:

- ❑ my_dataset.zip 包含影像檔案及 YOLO 標註檔
- ❑ x_obj.data 資料集設定檔
- ❑ x_obj.names 資料標籤名稱
- ❑ x_yolov4-tiny-custom.cfg 為自定義模型及參數
- ❑ x_train.txt 自定義資料訓練集檔案列表
- ❑ x_val.txt 自定義資料驗證集檔案列表
- ❑ *.jpg 測試用影像



AI Model 訓練過程

x_obj.data 資料集設定檔，classes 為有 五個類別
其餘設定為個資料的路徑

內容如下:

```
classes = 5  
train = data/x_train.txt  
valid = data/x_val.txt  
names = data/x_obj.names  
backup = /my_drive/x_yolov4-tiny
```

Ideas
Hatch

AI Model 訓練過程

x_obj.names 資料標籤名稱，設定各類別名稱。
內容如下：

Cerambycidae
Coccinellidae
Hydrophilidae
Lucanidae ●
Pentatomidae



Ideas
Hatch



AI Model 訓練過程

x_yolov4-tiny-custom.cfg 自定義模型及參數

這個參數檔是從 darknet/cfg/yolov4-tiny-custom.cfg 修改而來

line 6 : batch=64

line 7: subdivisions=1

line 8: width=416 # 為32的倍數

line 9: height=416 # 為32的倍數

line 20: max_batches=10000 # 為 classes x 2000 , 範例類別為 5

line 22: steps=8000,9000 # 為 max_batches 的 80% 與 90%

line 212,263: filter=30 # (classes + 5) x 3

line 220,269 : classes=5 #物件類別數量(手勢 :剪刀,石頭,布)

Ideas
Hatch



AI Model 訓練過程

x_train.txt 自定義資料訓練集檔案列表，為資料集的路徑及檔案名稱。

注意：最後一筆資料不要換行，訓練過程有可能會出現錯誤。

部分內容列舉如下：

data/my_dataset/train/001-415_jpg.rf.3c1b02750d5fcd75452749aec4a54bae.jpg
data/my_dataset/train/001-415_jpg.rf.747733229278482596b410be8b86703e.jpg
data/my_dataset/train/001-415_jpg.rf.99b1a5a384c15e73a0cb5e7a3c79f950.jpg
data/my_dataset/train/001-510_jpg.rf.615ee28a37452bcb5810cdf4f70be2c0.jpg
data/my_dataset/train/001-510_jpg.rf.71b179156454ad2525f38a26f3b6f3d2.jpg
data/my_dataset/train/001-510_jpg.rf.e82ced3ccf822712fc5918d43042ec51.jpg
data/my_dataset/train/004-78_jpg.rf.0c2c62e25cf83775ca4df74613e70e2c.jpg
data/my_dataset/train/004-78_jpg.rf.cf32405a379d09f958767bd311d81bf3.jpg
data/my_dataset/train/004-78_jpg.rf.f9b8c7c495f633240264cfdd147bfc11.jpg

.....



AI Model 訓練過程

x_val.txt 自定義資料驗證集檔案列表，同樣為資料集的路徑及檔案名稱。
注意路徑是需自建 my_dataset 資料夾，可以換取不同資料夾名稱。

部分內容列舉如下：

data/my_dataset/valid/011_jpg.rf.2735c79fba40e69d0b27dc4487e355f0.jpg

data/my_dataset/valid/35_jpg.rf.0f56854f0374efca955292cdbac0f65c.jpg

data/my_dataset/valid/422_jpg.rf.bb8d7600afd08bf2fc461e7a3867ae05.jpg

data/my_dataset/valid/448_jpg.rf.a3b989d7a036bd885ab9c68b1acb7124.jpg

data/my_dataset/valid/449_jpg.rf.d69ba29e9a5243744dab715a6def8976.jpg

data/my_dataset/valid/536_jpg.rf.b9e30d43b2eda705f2ccf6c3ef246320.jpg

.....

Google colab

- 不必進行任何設定
- 可開發及訓練類神經網路
- 可進行 **AI** 研究
- 輕鬆共用



AI Model 訓練過程

開啟 Google Colab 並新增建立記事本。

首先指令 `!nvidia-smi`，確認 GPU 有上線

```
[ ] !nvidia-smi

Thu May 18 11:11:16 2023

+-----+
| NVIDIA-SMI 525.85.12      Driver Version: 525.85.12      CUDA Version: 12.0     |
+-----+-----+
| GPU  Name          Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC  |
| Fan  Temp   Perf    Pwr:Usage/Cap|      Memory-Usage | GPU-Util  Compute M.  |
|                                       |                  |              MIG M.   |
+-----+-----+
|  0  Tesla T4             off      | 00000000:00:04.0 off  |          0             |
| N/A   55C    P8         10W / 70W   |  0MiB / 15360MiB |           0%          Default |
|                                       |                  |              N/A      |
+-----+-----+

Processes:
+-----+-----+
| GPU  GI  CI           PID   Type   Process name                      GPU Memory |
| ID   ID  ID                                   |             Usage          |
+-----+-----+
| No running processes found |
+-----+-----+
```

AI Model 訓練過程

連接 Google Drive 網路硬碟，並取名為 my_drive

```
[ ] from google.colab import drive
    drive.mount('/content/drive')

Mounted at /content/drive

[ ] !ln -s /content/drive/MyDrive/ /my_drive
```

Ideas
Hatch



AI Model 訓練過程

從 Github 下載 Darknet

!git clone <https://github.com/AlexeyAB/darknet>

```
[ ] !git clone https://github.com/AlexeyAB/darknet
```

Ideas
Hatch



AI Model 訓練過程

下載預設權重

!wget -N https://github.com/AlexeyAB/darknet/releases/download/darknet_yolo_v4_pre/yolov4-tiny.weights

```
[ ] !wget -N https://github.com/AlexeyAB/darknet/releases/download/darknet\_yolo\_v4\_pre/yolov4-tiny.weights
```

修改 Makefile

將 GPU, CUDNN, CUDNN_HALF, OPENCV 設定為 Enable

```
!sed -i 's/OPENCV=0/OPENCV=1/' Makefile
```

```
!sed -i 's/GPU=0/GPU=1/' Makefile
```

```
!sed -i 's/CUDNN=0/CUDNN=1/' Makefile
```

```
!sed -i 's/CUDNN_HALF=0/CUDNN_HALF=1/' Makefile
```

```
[ ] %cd darknet
```

```
/content/darknet
```

```
[ ] !sed -i 's/OPENCV=0/OPENCV=1/' Makefile
```

```
!sed -i 's/GPU=0/GPU=1/' Makefile
```

```
!sed -i 's/CUDNN=0/CUDNN=1/' Makefile
```

```
!sed -i 's/CUDNN_HALF=0/CUDNN_HALF=1/' Makefile
```

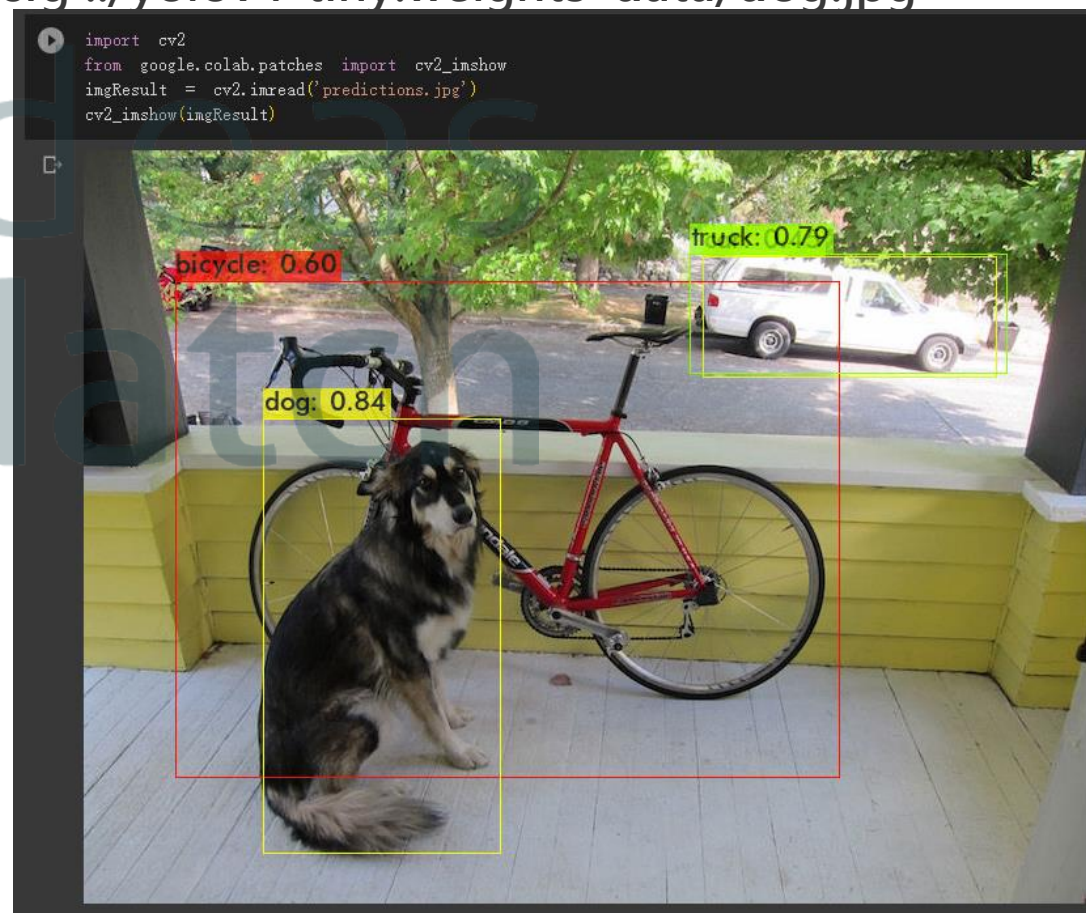

AI Model 訓練過程

測試 Darknet，使用內建的 coco 資料集

```
!./darknet detector test cfg/coco.data cfg/yolov4-tiny.cfg ../yolov4-tiny.weights data/dog.jpg
```

辨識完成後，編寫簡短的 python 來顯示結果
如左圖，即成功完成辨識。

```
import cv2
from google.colab.patches import cv2_imshow
imgResult = cv2.imread('predictions.jpg')
cv2_imshow(imgResult)
```





AI Model 訓練過程

回到 content 並下載 預權重 yolov4-tiny.conv.29

!wget -N https://github.com/AlexeyAB/darknet/releases/download/darknet_yolo_v4_pre/yolov4-tiny.conv.29

```
[ ] !wget -N https://github.com/AlexeyAB/darknet/releases/download/darknet_yolo_v4_pre/yolov4-tiny.conv.29
```

複製 x_yolov4-tiny-custom.cfg 至 darknet/cfg 內

!cp /my_drive/g_trindata/project1/g_yolov4-tiny-custom.cfg darknet/cfg/

```
[ ] !cp /my_drive/g_trindata/project6/x_yolov4-tiny-custom.cfg darknet/cfg/
```

AI Model 訓練過程

複製其餘資料集設定檔 至 darknet/data 內

```
!cp /my_drive/g_trindata/project6/x_obj.names darknet/data
```

```
!cp /my_drive/g_trindata/project6/x_obj.data darknet/data
```

```
!cp /my_drive/g_trindata/project6/x_train.txt darknet/data
```

```
!cp /my_drive/g_trindata/project6/x_val.txt darknet/data
```

```
[ ] !cp /my_drive/g_trindata/project6/x_obj.names darknet/data
```

```
[ ] !cp /my_drive/g_trindata/project6/x_obj.data darknet/data
```

```
[ ] !cp /my_drive/g_trindata/project6/x_train.txt darknet/data
```

```
[ ] !cp /my_drive/g_trindata/project6/x_val.txt darknet/data
```



AI Model 訓練過程

解開 資料集 zip 檔案並放入 darknet/data 內

```
!unzip /my_drive/g_trindata/project6/my_dataset.zip -d darknet/data
```

```
[ ] !unzip /my_drive/g_trindata/project6/my_dataset.zip -d darknet/data
```

Ideas
Hatch

AI Model 訓練過程

準備就緒,開始訓練

```
!./darknet detector train data/x_obj.data cfg/x_yolov4-tiny-custom.cfg ../yolov4-tiny.conv.29 -map -dont_show
```

```
[ ] !./darknet detector train data/x_obj.data cfg/x_yolov4-tiny-custom.cfg ../yolov4-tiny.conv.29 -map -dont_show
```

約莫訓練 1~2 小時左右，訓練成果會在預設資料夾 `x_yolov4-tiny` 內生成 `x_yolov4-tiny-custom_final.weights` 為最終的權重檔



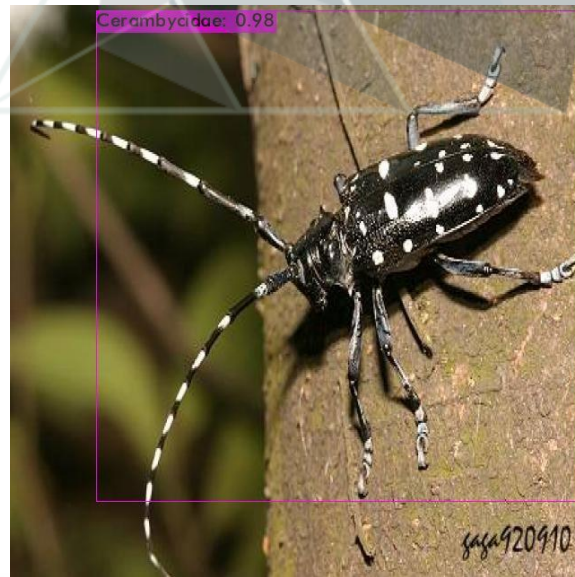
AI Model 訓練過程

測試驗證訓練成果，透過以下辨識指令來辨識

```
!./darknet detector test data/x_obj.data cfg/x_yolov4-tiny-custom.cfg ../x_yolov4-tiny-custom_final.weights ../darknet/data/my_dataset/test/test.jpg
```

```
[1] !./darknet detector test data/x_obj.data cfg/x_yolov4-tiny-custom.cfg ../x_yolov4-tiny-custom_final.weights ../darknet/data/my_dataset/test/test.jpg
```

AI Model 訓練過程



AI Model 訓練過程

在與 HUB8735 Ultra 應用之前，必須透過 Realtek 專責網頁先行轉檔。

將 權重檔(g_yolov4-tiny-custom_final.weights 及 x_yolov4-tiny-custom.cfg 檔案，使用 zip 壓縮打包。

開啟 <https://www.amebaiot.com/zh/amebapro2-ai-convert-model/> 網頁，填入 zip 檔案，及選擇任一張驗證圖片並上傳。

Upload zip file including a cfg file and a weights file(required, please upload the folder or compressed file contained the ".cfg" and ".weights" files, all named in English, limit:35MB)

選擇檔案 未選擇任何檔案

Upload one jpg file (required, limit:1MB)

選擇檔案 未選擇任何檔案



AI Model 訓練過程

約莫 1 ~ 5 分鐘，會自動回覆至登記註冊 e-mail 信箱裡。並將 nb 檔案下載。

開啟檔案總管至

C:\Users\\AppData\Local\Arduino15\packages\ideasHatch\hardware\AmebaPro2\version\variants\common_nn_models

並更名置換 yolov4_tiny.nb 檔案。

Ideas
Hatch



程式設計

Ideas Hatch

利用範例 ObjectDetectionCallBack 修改及新增功能
首先將 ObjectClassList.h 設定 5 個名稱

```
// List of objects the pre-trained model is capable of recognizing
// Index number is fixed and hard-coded from training
// Set the filter value to 0 to ignore any recognized objects
ObjectDetectionItem itemList[5] = {
    {0, "Cerambycidae", 1},
    {1, "Coccinellidae", 1},
    {2, "Hydrophilidae", 1},
    {3, "Lucanidae", 1},
    {4, "Pentatomidae", 1},
};
```

AI Model 訓練過程

五種不同辨識類別，設定不同的顏色標註

```
// Draw boundary box
printf("Item %d %s:\t%d %d %d %d\n\r", i, itemList[obj_type].objectName, xmin, xmax, ymin, ymax);
switch(obj_type){
  case 0:
    OSD.drawRect(CHANNEL, xmin, ymin, xmax, ymax, 3, OSD_COLOR_WHITE);
    break;

  case 1:
    OSD.drawRect(CHANNEL, xmin, ymin, xmax, ymax, 3, OSD_COLOR_RED);
    break;

  case 2:
    OSD.drawRect(CHANNEL, xmin, ymin, xmax, ymax, 3, OSD_COLOR_BLUE);
    break;

  case 3:
    OSD.drawRect(CHANNEL, xmin, ymin, xmax, ymax, 3, OSD_COLOR_GREEN);
    break;

  case 4:
    OSD.drawRect(CHANNEL, xmin, ymin, xmax, ymax, 3, OSD_COLOR_MAGENTA);
    break;
}
```

AI Model 訓練過程

將程式上傳至 HUB8735 Ultra，以下是 Video stream 接收至 VLC 軟體來驗證辨識能力

