

聯網智造基地





0



U B

國產IC應用實作開發系列 HUB 8735 智慧硬幣辨識

講師: 章育銘

112/06/1



課程檔案





https://github.com/wildman8606/HUB-8735-Coin-Detection-tutorial

網址:



講師: 章育銘

退役競賽獎金獵人 #門薩選委會監事 # 資訊工業策進會 講師 #工業技術研究院副工程師 #工業局物聯網策略解決專家 #甲種電匠 # 交大電控博士休學中 ... +10



|賽暨頒獎典禮

中華電信 Chunghwa Telecom



\$200,000



• 相關教學

WE-I Plus智慧儀錶辨識:

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

[MQTT 與 AloT 整合運用 社群分享會]CoreMaker-01 串接 Arduino 環境AI 語音辨識&手勢判斷 教學: https://www.youtube.com/watch?v=dMBl8v92u_4&t=1050s











1. 前言 2. 以Arduino IDE操作Smart-CAM 3. labelImg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 Smart AI CAM

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

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

前言

101

·以Arduino IDE操作Smart-CAM -前言



Standard SDK **Development**



AT Commands

透過GCC building on Linux or windows environment • 提供Edge AI、AWS-IoT · Amazon Kinesis Video streams的應用範例

透過Arduino平台, 提供 HUB 8735 各 種功能範例,也可透 過Github 同步原廠 的Release。

無須進行編譯。 透過外部Serial Port 下AT Commands指 令控制。

優點:

缺點:

•

•

•

程式修改掌握度高。

安裝環境較複雜。

需原廠提供SDK資源。

可以處理更複雜的工作程序。

適合底層或熟悉各介面的開發人員。

編譯& 燒錄時間較長且較繁瑣。

、以Arduino IDE操作Smart-CAM -前言

HUB 8735 工具包

https://github.com/ideashatch/HUB-8735/tree/main/HUB%208735%E5%B7%A5%E5%85%B7%E5%8C%85

Standard SDK Linux Toolchain 安裝參考

https://github.com/ambiot/ambpro2_sdk

MinGW & MSYS2安裝參考

https://opensourcedoc.com/windows-programming/mingw-msys/

Edge AI Demo Start 參考網站

https://github.com/ambiot/ambpro2_sdk/blob/main/doc/NN_example_README.md

ASDK 10.3.0 for windows

https://www.dropbox.com/s/bazelvm65zx6zlq/asdk-10.3.0-mingw32-newlib-build-3633-x86_64.zip?dl=0

CMAKE 3.20.0下載

https://github.com/Kitware/CMake/releases/download/v3.20.0-rc1/cmake-3.20.0-rc1-windows-x86_64.msi



、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

辺別圏太人貧訊上美束進智

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。

sketch_may	/09a Arduino 1.8.		自動格式化 お存草稿碼 たてはF#T たてはF#T
		偏好設定 ×	sketch_may09a IV/LIII的业里有較大
檔案 編輯 草種	高碼 工具 說明 🛛 🛛	all de la mais	
		設定 網路	VOID SE 序列埠監控視台 Ctrl+Shitt+M
新增	Ctrl+N	草稿碼灣的位置:	// pu ^{序列編国家 Ctrl+Shift+L} to run once:
開啟	Ctrl+O	C:\Users\Weko\Documents\Arduino	WiFi101 / WiFiNINA Firmware Updater
開啟最近	>	編輯器語言: System Default 〈 (需要重新啟動 Arduino)	開設版: "Ameba RIL8195A" Step.3 、 開設版管理員 序列埠 Step.3 、 Ameba Antor (s2-roits) Boards >
		編輯器字型大小: 20	取得開發板資訊 AmebaD ARM (32-bits) Boards >
阜樯傿薄	>		Arduino AVR Boards >
会告 /万 日			Void 10 周期語 STM32F1 Boards (Arduino_STM32) >
単じ17リ	-	Theme: Default theme ~ (需要重新啟動 Arduino)	WE-I Plus Boards
關閉	Ctrl+W	▶ 顯示詳細輸出: □ 編譯 □ 上傳	// put vour main code here, to run reperent.
		▲禮器警告: 毎	图2 淮 λ ' 閉發板管理昌'
儲存	Ctrl+S		圖J. 進入 開發 版目 生貞
中大蛇横	Ctel Chift C	□ 與示行數 □ 取用程式碼指叠功能	
力1分利1面	. Cur+shint+s	□ 上傳後驗證程式碼	
		✓ 図 啟動時檢查有無更新 Step.2 ✓ 啟動時檢查有無更新 Step.2 ✓ 驗證或上傳時先存檔	類型 全部 🗸 Realtek Ameba Boards
頁面設定	Ctrl+Shift+P		Realtek Ameba Boards (32-bit Arm v8M @500MHz)
Cton 1			by Realtek
Step.1	Ctrl+P	観外的開發被管理員網址: deashatch/HUB-8735/raw/mann/amebapro2_ardumo/Ardumo_package/ideasHatch.json L	此銀行地営力期間移物: AMB82 (RTL9735B), AMB82-MINI (RTL8735B), Ctan 人
		在偏好設定檔裡還有更多設定但可且按調理	
盾収む中	Ctrl L Commo	C:\Users\Weko\AppData\Local\Arduino15\preferences.txt	More Into
	CurrComma	(只能在Arduino未執行之時進行編輯)	4.0.0-build20221125 Y 安装
			Realtek Ameba Boards (32-bits ARM Cortex-M3)
離開	Ctrl+Q	確定 取消	+10.3-900720205000 by Realtek 版本2.0.6 INSTALLED 4 10.3-bmi20230316
	•		此素件包含的服装板: 4.0.3-build20230308 4.0.3-build20230308
周1 谁λ	'偏好铅定'	周2 美入蓮結	国Λ 龄λ' Pooltok '比别問発板安奘
			圖4. 期八 Reditek 找到用發版女衣

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

sketch_	mayrua	Arduino 1.8.19	
當案 編輯	草稿碼	工具 説明	
90	B 1	自動格式化 Ctrl+ 封存草稿碼	
sketch_	may10a	修正編碼並重新載入 管理程式庫 Ctrl+:	Shift+1
//	put	序列埠監控視函 Ctrl+: 序列繪圖家 Ctrl+:	Shift+M Shift+L pnce:
}		WiFi101 / WiFiNINA Firmware Updater 	Step.1
void	looj	* Erase All Flash Memory (16MB): "Disable" * Auto Flash Mode: "Disable" * Standard Lib: "Disable"	AmebaPro2 ARM (32-bits) Boards - ideasHatch HUB-8735 Arduino AVR Boards
//	put	* Upload Speed: "921600" 序列埠 取得開發板資訊	'epeatedly:
ĵ		燒錄器 燒錄Bootloader	>

圖1.選擇'HUB-8735'開發板設定

個茶 補助 早価価 上央 説明	COM5 - □ ×
VideoOnly §	Interlace V IS Initialized
#include "StreamIO.n"	interface 1 is initialized
<pre>#include "VideoStream.h"</pre>	cfg size lib = 27, cfg size user = 27
<pre>#include "RTSP.h"</pre>	
	Initializing WIFI
#define CHANNEL 1	[Driver]: [HALMAC]
	11692M HALMAC_MAJOR_VER = 1
// Default preset configurations for each video chann	hel HALMAC_PROTOTYPE_VER = 4
// Channel 0 : 1920 x 1080 30FPS H264	HALMAC_MINOR_VER = 20
// Channel 1 : 1280 x 720 30FPS H264	HALMAC_PATCH_VER = - download_firmware_88xx_HALMAC_RET_SUCCES
// Channel 2 : 1280 x 720 30FPS MJPEG	RFE type = 0
,,	start_addr=(0x4000), end_addr=(0x8000), buffer_size=(0x4000),
VideoSetting config(CHANNEL):	
BTSP rtsn:	WIFI initialized
StreamIO videoStreamer(1 1): $(/ 1 \text{ Input Video} \rightarrow 1)$	Ctop 2
Screamic VideoScreamer(1, 17, 77 1 input video > 1	[Driver]: set ssid [KuroNeko]
chan agid() = "WaneNeke", (/ your potyonk CCTD (no	
char ssid[] = Kuroneko; // your network SSID (na	
char pass[] = "2/182818"; // your network passw	VOI (Hallingan Conow innestamp (NL(newinne) (17200 band) Clear output
(
草稿碼使用了 4239360 bytes (25%) 的程式儲存空間。上限為 167	777216 bytes •
Enter Flash Mode!	
Start Upload Flash	
Unloading unload su	

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)

💿 sketch_may09a | Arduino 1.8.19 横安 绝起 苔枝硬 工具 鉛明

0

新增 Ctrl+N 開啟 Ctrl+O 開啟最近 草稿碼簿	,	<i>Ş</i>	
範例 關閉 Ctrl+W 儲存 Ctrl+S 另存新檔 Ctrl+Shift+S	09.USB 10.StarterKit_BasicKit 11.ArduinoISP	here, to run once:	
頁面設定 Ctrl+Shift+P 列印 Ctrl+P 偏好設定 Ctrl+Comma 離開 Ctrl+Q	任何板子皆可用的範例 Adafruit Circuit Playgro Bridge Ethernet Firmata	pund > > >	E
<pre>// put you }</pre>	LiquidCrystal SD Stepper Temboo TFT	<pre, pre="" repeatedly<="" run="" to=""></pre,>	
	HUB-8735的範例 AmebaAnalog AmebaBLE AmebaFileSystem AmebaGPIO AmebaGTimer AmebaHttp AmebaMOTTClient	> > > > >	
	AmebaMultimedia AmebaNN AmebaNotify AmebaPowerMode	 Audio CaptureJPEG MotionDetection RecordMP4 	
	AmebaRTC AmebaSDDownload AmebaSPI	StreamRTSP DoubleVideo DoubleVideoWithAudio SingleVideoWithAudio	
	AmebaWatchdog AmebaWire WiFi ▽	> VideoOnly >	

or	onCallback Arduino 1.8.19			
ŧ	工具	說明		
ł		自動格式化	Ctrl+T	
		封存草稿碼		
n		修正編碼並重新載入		
	1	管理程式庫	Ctrl+Shift+I	
		序列埠監控視窗	Ctrl+Shift+M	
	, 1	序列繪圖家	Ctrl+Shift+L	
Э		WiFi101 / WiFiNINA Firmware Updater		
1		開發板: "HUB-8735"	>	-amb82-m
		* Erase All Flash Memory (16MB): "Disable"	>	
		* Auto Flash Mode: "Disable"	>	
2		* Standard Lib: "Disable"	>	tter vid
		* Upload Speed: "2000000"	>	921600
		序列埠	>	• 2000000
1		取得開發板資訊	0	230400
]	1	燒錄器	>	dels usi
Э		傳錄Bootloader		necessa

圖2. 傳輸速度選擇

圖1. 範例'VideoOnly'

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





圖1~3. 各類USB to TTL

10

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

HUB 8735





0

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



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

步驟1. 下載VLC Media Player:

0

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跳腺 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)



With a state of the s

 ◎ VideoOnly Arduino 1.8.19 檔塞 编輯 普茲碼 工具 範囲 	🛓 VLC 煤體播放器			
	<mark>媒體(M)</mark> 播放(P) 音訊(A) 視	訊(V) 字幕(T) 工具(S) 檢視(V) 說明(H)	📥 開啟媒體	- ×
	▶ 開啟檔案(F)	Ctrl+O		
THEITOT SET USET GALA SIZE TOT, NEED MOTE CHEM .	▶ 開啟多個個業(U)	Ctrl+E	🕞 檔案(F) 🛛 📎 光碟(D) 🛛 🚼 網	1路(N) 📑 摘取裝置(D)
hal voe send2voe too long 51179 cmd 0x0000020				
#	🚏 開啟網路串流(N)	Ctrl+N	網路通訊協定	
- Summary of Streaming - Step.2	(基) 用歐旗取表重(C) 從剪貼簿開啟位置(L) 開啟最近使用的媒體(R)	Ctrl+V	請輸入網址: Step.3	
[#] Channel: 1	儲存播放清單至檔案(F)	Ctrl+Y	rtsp://192.168.1.18:554	X
Encoder type: H264	轉換(R) / 儲存	Ctrl+R	http://www.example.com/stream.avi	
Resolution: VIDEO HD	(**) 串流(S)	Ctrl+S	rtp://@:1234	
Video width: 1280	到達播放清單結尾時離開		mms://mms.examples.com/stream.asx	
/Video height: 720	□ 離用(Q)	CIII+Q	rtsp://server.example.org:8080/test.sdp	
//fps: 30			http://www.yourtube.com/watch?v=ggo4x	
bps: 4194304 Cton 1				
R1 - RTSP - Strtsp://192.168.1.18:554				
ch ch		MJE OR	☑ 顯示更多選項(M. Step.4	
☑ 自動捲動 □ Show timestamp NL(newline)			仲取 1000 高秋 🛋	問始時間 00H·00m·00≈000 €
x				
	-			停止時間 00H:00m:00s.000 🖨
早崎崎使用」 4239360 bytes (25%) 的栓式储存空间。上限			□ 同乎採放史→研棋體/旗队份空訊&	
Enter Hiash Mode:				毎5天 ´ …)
Start Upload Flash			MRL rtsp://192.168.1.18:554	
Uploadinguplo		XI		
End Upload Flash			編輯選項 :network-caching=1000	
<				
83				播放(P) ▼ 取消(C)
🕂 🔎 在這裡輸入文字來搜 🧯 🔄 🖸 💿 📰	💿 🤹 🛓			

圖1. VLC Media Player操作



10T

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

Itsp://192.168.1.18:554 - VLC 媒體播放器

- 🗆 🗙



圖1. VLC Media Player操作

labellmg標注工具介紹-讓AI知道這是什麼?



10T

101









7(7)



8 (8)





1(1)

1(1)



圖2. 自動算錢



2 (2)





7 (7)

圖1. MNIST



₩三、labellmg標注工具介紹 –蒐集照片

s a	
HTTPDisplayJPEG Arduino 1.8.19	- 0 X
HTTPDisplayJPEG §	
/* This example uses the camera to capture a JPEG image	start_addr=(0x4000), end_addr=(0x8000), buffer_size=(0x4000), smp_^
* and sends the image to a browser using HTTP.	
* Use any prowser and connect to the board's IP addre	^e WIFI initialized
"/ #include ≪WiFi b	
<pre>#include "VideoStream.h"</pre>	[Driver]: set ssid [KuroNeko]
	[Driver]: OpBeacon rtw get sec in (inlen: 285 rsn len: 0 what ler
#define CHANNEL 0	[Driver]: start auth to 54:04:a6:5c:db:54
// Use a pre-defined resolution, or choose to configure	[Driver]: auth success, start assoc
<pre>VideoSetting config(VIDEO_FHD, CAM_FPS, VIDEO_JPEG, 1);</pre>	
<pre>//VideoSetting config(1072, 1072, CAM_FPS, VIDEO_JPEG,</pre>	[Driver]: association success(res=3)
<pre>char ssid[] = "KuroNeko"; // your network SSID (name</pre>	<pre>P[Driver]: wlan0: DL RSVD page success! DLBcnCount:1, poll:1</pre>
char pass[] = "2/182818"; // your network passwor	
WiFiServer server (80) ·	[Driver]: set pairwise key to hw: alg:4(WEP40-1 WEP104-5 TKIP-2 AF
<	
上傳完單。	[Driver]: set group key to nw: alg:2(WEP40-1 WEP104-5 IKIP-2 AES-4
早禍疇便用	Interface 0 IP address : 192,168,1.5
Enter Flash Mode!	
Start Upload Flash	
Uploadingupload success	☑ 自動捲動 □ Show timestamp
Ena upioad Flash	
<	
15	HUB-8735, Disable, Disable, Disable, 200000 ½ COM4
- 王 👂 在這裡輸入文字來搜 🥂 🕎 📮 🔍 💿 📥	^ ≌ @ ↓) ⊗ ^{p+0/30} □
圖1. 開啟HTTP顯示	rJPEG範例

5 B 27

AICE

1 O T

、labellmg標注工具介紹 –操作軟體



10T

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





「「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 'RAWdata'資料夾及上傳資料

1 0 T

0

101

使用Google Colab製作AI模型-前置準備

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

上傳2個檔案 Data_Augmentation_CLODSA.ipynb Train_YOLO_Object_Detection_With_Colab.ipynb



四、使用Google Colab製作AI模型-資料前處理1



10 T

0

101

一四、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

10T

101

一四、使用Google Colab製作AI模型-資料前處理2



问述。使用Google Colab製作AI模型-設定GPU運算模式



、使用Google Colab製作AI模型-跑YOLOv4-tiny模型

CO 4 ai_0514.ipynb ☆ 地致 短期 1940年前 版入 92-7669 丁目 1988	▲ ai_0514.ipynb ☆ 地変 編編 Ma相単布 語入 執行部69 工具 2081 戸6829所行的第一
·····································	
	Step 3: Download pre-trained VOLOV4 weights
 Step 2: Cloping and Building Darknet 	
	(x) YOLOV4 has been trained already on the coco dataset which has 80 classes that it can predict. We will grab these pretrained weights so that we can run YOLOV4 on these pretrained classes and get detections.
The following cells will clone darknet from AlexeyAB's famous repository, adjust the Makefile to enable OPENCV and GPU for darknet and then build darknet.	
Do not worry about any warnings when you run the 'Imake' cell!	[5] !wget <u>https://github.com/AlexeyAB/darknet/releases/download/darknet_rolo_v4_pre/yolov4-tiny.conv.29</u> 10
	!wget https://github.com/AlexeyAB/darknet/releases/download/darknet_yolo_v4_pre/yolov4-tiny.weights
2 lgit clone <u>https://github.com/AlexeyAB/darkmet</u>	2023-05-13 16:02:08 https://github.com/likewy84 darkmet/releases/download/darkmet_volo_v4_pre/volov4-tinv.comv.29 Resolving github.com (github.com) 140.82:121.3 Comparison to github.com (github.com) 202.121.3):443 Comparison to github.com (github.com) 202.121.3):443
C. Cloning into 'darknet' remote: Enumerating objects: 15521, done.	HTTP request sent, avaiting response 302 Found Location: https://objects.githubusecontent.com/github-production-release-asset-2e65be/75388965/28807d00-3ea4-11eb-97b5-4e846ecd1d057X-Am
remote: Counting objects: 100% (7/7), done. remote: Compressing objects: 100% (7/7), done.	2023-05-13 16:52:08 https://objects.githubusercontent.com/github-production-release-asset-2e65be/75388965/28807d00-3ea4-11eb-97b5-4c Resolving objects.githubusercontent.com (objects.githubusercontent.com) 185. 199. 108. 133, 185. 199. 109. 133, 185. 199. 110. 133,
remote: Total 15521 (delta 0), reused 5 (delta 0), pack∽reused 15514 Receiving objects: 100% (15521/15521), 14.19 MiB 19.80 MiB/s, done.	Connecting to objects.githubusercontent.com (objects.githubusercontent.com) [185, 199, 108, 133]:443 connected. HTTP request sent, awaiting response 200 OK
Resolving deltas: 100% (10412/10412), done.	Length: 19789716 (19M) [application/octet-stream] Saving to: 'yolov4-tiny.conv.29'
✓ [2] # change makefile to have GPU and OPENCV enabled	yolov4-tiny.conv.29 100%[=====>] 18.87M 31.0ME/s in 0.6s
<pre>bot defines lsed =i 's/OPENCY=0/OPENCY=1/ Makefile lsed =i 's/CPUEn/OPEnt/' Makefile</pre>	2023-05-13 16:52:09 (31.0 MB/s) = 'yolov4-tiny.conv.29' saved [19789716/19789716]
○ Ised -1 S/OUN=0/CUDN=1/ Ised -1 S/CUDN=0/CUDN=1/ Makefile	2023-05-13 16:52:09 <u>https://github.com/AlexevAB/darkmet/releases/download/darkmet_volo_v4_pre/volov4-tipv_weights</u> Resolving github.com (github.com) 140.82.121.3
Iseu -1 S/CONS_MLC=0/CONS_DUC=1/ MARTILE	Connecting to github.com (github.com) 140.82.121.3 :443 connected. HTTP request sent, awaiting response 302 Found
	Location: https://objects.githubusercontent.com/github-production=release=asset=2e65be/75389865/2280e00-3ead=11eb-8e80-28d71659f56c7X-in 2022-06-13_16:52:00https://objects.githubusercontent.com/githubusercontent_asset=2e65be/75389865/2280e00-3ead=11eb-8e80-28d71659f56c7X-in 2022-06-13_16:52:00https://objects.githubusercontent.com/githubusercontent/asset=2e65be/75389865/2280e00-3ead=11eb-8e80-28d71659f56c7X-in 2022-06-13_16:52:00https://objects.githubusercontent.com/githubusercontent/asset=2e65be/75389865/2280e00-3ead=11eb-8e80-28d71659f56c7X-in 2022-06-13_16:52:00https://objects.githubusercontent.com/githubusercontent/asset=2e65be/75389865/2280e00-3ead=11eb-8e80-28d71659f56c7X-in 2022-06-13_16:52:00https://objects.githubusercontent.com/githubusercontent/asset=2e65be/75389865/2280e00-3ead=11eb-8e80-28d71659f56c7X-in 2022-06-13_16:52:00https://objects.githubusercontent.com/githubusercontent/asset=2e65be/75389865/2280e00-3ead=11eb-8e80-28d71659f56c7X-in 2022-06-13_16:52:00https://objects.githubusercontent.com/githubusercontent/asset=2e65be/75389867280e00-3ead=11eb-8e80-28d71659f56c7X-in 2022-06-13_16:52:00https://objects.githubusercontent/asset=2e65be/75389667280e00-3ead=11eb-8e80-28d71659
0 (b) 70xxxx101101. 72x10112.32	0 7 元成时间,反应12.32
CO ⁴ ai_0514.ipynb ☆ 描案 編輯 特得書面 插入 载行翻發 丁耳 1988 已结存所有算更	CO Lai_0514.ipynb ☆ 施裁 編輯 檢授書單 加入 執行總成 工具 装明 已就在所有鉴照
▲ ai_0514.ipynb ☆ 檔案 編輯 檢視畫面 插入 執行階段 工具 說明 已儲存所有重更 + 程式碼 + 农字	● ai_0514.jpynb ☆ > 接表 编辑 协说集署 描入 執行階段 工具 款時 已经存折合股里 = + 程式碼 + 文字
CO ▲ ai_0514.ipynb ☆ 檔案 編輯 檢視畫面 插入 執行階段 工具 說明 已結存所有鉴更 IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII	$\begin{array}{c} & \overbrace{al_{0}}^{\bullet} \text{S014.ipynb} \ \widehat{\alpha} \\ & \overbrace{bask} with break with break in A with Rel II with Eddering with the second $
○○● ai_0514.jpynb ☆ 檀本 編輯 機構畫面 插入 執行階段 工具 訪明 已結存所指量更 □□ + 程式碼 + 文字 □□ This will output the image with the detections shown. The most recent detections are always saved to 'predictions.jpg'	Co al_0514.jpynb ☆ 接接 編輯 檢視業業 編入 執行指於 工具 装持 已結开所省物里 干燥式的 + 文方 文 5 (0) # down insee wing our holper function indew('predictions.jpg') (x)
 CO ▲ ai_0514.jpynb ☆ 檔案 編輯 檢視畫面 插入 執行階段 工員 說明 已結存所有重更 二 + 程式碼 + 文字 □ This will output the image with the detections shown. The most recent detections are always saved to 'predictions.jpg' (x) Try out the examples below for yourself! 	Co
 CO ▲ ai_0514.ipynb ☆ 標本 編輯 檢視畫面 插入 執行階段 工員 說明 已結存所指整更 二 + 程式碼 + 文字 □ This will output the image with the detections shown. The most recent detections are always saved to 'predictions.jpg' (x) Try out the examples below for yourself! □ = run darknet detection on test images 	▲ ai_0514.ipynb ☆ 版表 編輯 物理義業 編入 執行指定 工具 読明 已建进所由型型 H 新聞 物理義業 編入 執行指定 工具 読明 已建进所由型型 H 新聞 物理 新聞 和 和 和 和 和 和 和 和 和 和 和 和 和 和 和 和 和 和
 CO ▲ ai_O514.ipynb ☆ 標本 編輯 檢視畫面 插入 執行階段 工員 說明 已結存所有鉴更 半程式碼 + 文字 This will output the image with the detections shown. The most recent detections are always saved to 'predictions.jpg' (x) Try out the examples below for yourself! ○ = run darknet detection on test images 1./darknet detector test cfg/coco.data cfg/yolov4-tiny.cfg yolov4-tiny.weights data/person.jpg 	Constraint index in the set of the set
 CO ▲ ai_O514.ipynb ☆ 標本 編輯 檢視識面 插入 執行階段 工具 說明 已結存所有望更 # 程式碼 + 文字 This will output the image with the detections shown. The most recent detections are always saved to 'predictions.jpg' (x) Try out the examples below for yourself! ● = run darknet detection on test images 1./darknet detector test cfg/cocc.data cfg/yolov4-tiny.efg yolov4-tiny.weights data/person.jpg CUDA-version: 11080 (12000), euDNN: 8.7.0, CUDNN_HALF=1, GPU count: 1 CUDN_HALF=1 	Asi_0514.jpynb ☆ Mak Mel Modern HA Minike Is and Eddernamme Hettis +2≠ Q 5 B f date integ units our helper function inder(prediction.jp?) H
Constant of the second	Constraint index An index Hait will be base mint with the line of the index Hait will be base mint our helper function index's production.jp?
 CO ▲ ai_O514.ipynb ☆ 標本 編輯 後根識面 插入 執行階段 工具 説明 已結存近白聖更 # 程式碼 + 文字 This will output the image with the detections shown. The most recent detections are always saved to 'predictions.jpg' (x) Try out the examples below for yourself! ● = run darkmet detection on test images 1./darkmet detector test cfg/coco.data cfg/yolov4-tiny.cfg yolov4-tiny.weights data/person.jpg CDN-yesion: 11080 (12000), cuDNN: 8.7.0, CUDN_HALF=1, GPU count: 1 CUDN_HALF=1 O = compute_apability = 760, cudnn_half = 1, GPU: Tesla T4 met.optimized_memory = 0 min_batch = 1, batch = 1, time_steps = 1, train = 0 	Asi_0514.kpynb ☆ Hat seis indext in A sinder II and Cathodram Hat seis indext in A s
 CO ▲ ai_OS14.ipynb ☆ 描葉 編輯 後根集面 插入 執行階段 工具 說明 已結存所有望更 ■ + 程式碼 + 文字 ■ This will output the image with the detections shown. The most recent detections are always saved to 'predictions.jpg' (x) Try out the examples below for yourself! ● = run darknet detection on test images 1./darknet detector test ofg/coco.data ofg/yolov4-tiny.ofg yolov4-tiny.weights data/person.jpg ● CDA+version: 11080 (12000), cuDN: S.7.0, CUDN_BALF=1, GPU count: 1 CUDN_HALF=1 OpenCV version: 4.2.0 0 : compute_sambility = 750, cudnn_half = 1, GPU: Tesla T4 met.optimized_sempory = 0 init_Datath = 1, bath = 1, time_steps = 1, train = 0 layer filters size/strd(dil) input output 0 trainet CUT_Time 	<pre></pre>
 CO ▲ ai_O514.ipynb ☆ 描葉 編輯 快機環面 插入 執行階段 工具 認明 已結存所有望更 ■ + 程式碼 + 文字 ■ This will output the image with the detections shown. The most recent detections are always saved to 'predictions.jpg' (x) Try out the examples below for yourself! ● = run darknet detection on test images 1./darknet detector test cfg/coco.data cfg/yolov4-tiny.cfg yolov4-tiny.weights data/person.jpg ● CDA-version: 11080 (12000), cuBN: 8.7.0, CDBN_BLEF1, GPU count: 1 OpenCV version: 4.2.0 0 : compute.agashilty = 750, cudm_half = 1, GPU: Tesla T4 met.optimet.genery = 0 mini_blatch = 1, time_steps = 1, train = 0 layer filters size/cit(dil) input output 0 Create (DDA-ters in = 0) input = 0 cupute.genery = 0 data 1.2 2, 208 x 208 x 32 0.075 BF loow = 64 = 33 x 3/2 2, 208 x 208 x 32 0.075 BF 	<pre>kij bij kijvyh * kij kij kij kij kij kij kij kij kij kij</pre>
 CO ▲ ai_O514.ipynb ☆ 描述 編輯 快程識 插入 執行階段 工具 10時 已結存所有望更 ■ + 程式碼 + 文字 □ + 昭式碼 + 文字 □ Try out the examples below for yourself! ○ = run darknet detection on test images 1./darknet detector test cfg/coco.data cfg/yolov4-tiny.cfg yolov4-tiny.weights data/person.jpg □ CDA-version: 11080 (12000), cuBN: 8.7.0, CUBN_HALF=1, GPU count: 1 CUBN_HALF=1 0 pentfV version: 4.2.0 0 : compute_acapability = 750, cudn_half = 1, GPU: Tesla T4 mini_batch = 1, time_steps = 1, train = 0 layer filters size/strid(dil) imput output 0 Create cUDA-strid(dil) imput output 0 Create CUDA-strid(dil) imput output 0 Create CUDA-strid(dil) imput 0 at 104 x 64 0.399 EP 2 conv 64 3 x 3/1 104 x 104 x 64 - 2104 x 104 x 64 0.399 EP 2 conv 64 3 x 3/1 104 x 104 x 64 - 2104 x 104 x 64 0.399 EP 	<pre>kij contraction contracti</pre>
 CO ▲ ai_O514.ipynb ☆ 描述 編輯 快程識 插入 執行階段 工具 10時 已经存进行管理 # HETE + 文字 This will output the image with the detections shown. The most recent detections are always saved to 'predictions.jpg' (x) Try out the examples below for yourself! I = run darknet detector on test images L/darknet detector test efg/coco.data efg/yolov4-tiny.efg yolov4-tiny.weights data/person.jpg CDA-version: 11080 (12000), cuDN: 8.7.0, CUDN_HALF=1, GPU count: 1 OpenCV version: 4.2.0 0 : compute_capability = 750, cudm_half = 1, GPU: Tesla T4 net.ortimized.memory = 0 mini_batch = 1, time_steps = 1, train = 0 layer filters size/strid(dil) input output Ocreate CDA-version: 108 x 3.2 2 008 x 202 x 108 x 32 0.075 BF lconv 64 3 x 3/1 104 x 104 x 4 -> 104 x 104 x 64 0.399 BF 2 conv 64 3 x 3/1 104 x 104 x 22 -> 104 x 104 x 104 x 22 to 199 BF sconv 32 3 x 3/1 104 x 104 x 32 -> 104 x 104 x 32 0.199 BF 	<pre>A si_0514.ipynb * isa sets tedage itA withink II a two Catendrates + With + 2* 0 f (B) U due inse wins out helper faction isaber's reference.jp() </pre>
 ● ai_0514.ipynb ☆ 描葉 編輯 快程識 插入 執行階段 工具 10時 已結存所有筆更 ■ + 程式碼 + 文字 □ + 程式碼 + 文字 □ * und arknet detection on test images 1./darknet detector test cfg/occo.data cfg/yolov4-tiny.efg yolov4-tiny.weights data/person.jpg (x) * run darknet detector on test images 1./darknet detector test cfg/occo.data cfg/yolov4-tiny.efg yolov4-tiny.weights data/person.jpg C CDA-version: 11080 (12000), cuDNN: 8.7.0, CUDNN_HALF=1, GPU count: 1 OpenCV version: 14.2.0 0 : compute_capability = 750, cudm_half = 1, GPU: Tesla T4 net.optimized_memory = 0 mini_bitcht = 1, time_steps = 1, train = 0 layer filters size/strid(dil) input output 0 Create cUDA-strama - 0 Create cUDA-stram	<pre>A si_0514.kpmb * the and bodge into an know into an</pre>
 ● ai_0514.ipynb ☆ 標本 編輯 後親重面 描入 執行期段 工員 説明 已結存所引望更 + 程式碼 + 文字 ■ This will output the image with the detections shown. The most recent detections are always saved to 'predictions.jpg' (x) Try out the examples below for yourself! ● = run darknet detection on test images 1./darknet detector test ofg/occo.data ofg/yolov4-tiny.efg yolov4-tiny.weights data/person.jpg ● CUDA-version: 11080 (12000), cuDNN: 8.7.0, CUDN_HALF=1, GPU count: 1 CUDN_HALF=1 OpenCV version: 4.2.0 0 : compute_camability = 750, cudm_half = 1, GPU: Tesla T4 net.optimized_memory = 0 min_ibatch = 1, batch = 1, time_steps = 1, train = 0 layer filters size/strd(dil) input output 0 Create cudm-handle 0 conv 32 3 x 3/2 406 x 416 x 3 -> 206 x 208 x 32 0.075 BF 1 conv 64 3 x 3/2 104 x 104 x 4 -> 104 x 104 x 32 0.199 BF 3 conte 5 4 1 conv 5 4 1 x 1/1 104 x 104 x 32 -> 104 x 104 x 32 0.199 BF 6 conte 5 4 1 conv 6 4 1 x 1/1 104 x 104 x 32 -> 104 x 104 x 32 0.199 BF 6 conte 5 4 1 conv 6 4 1 x 1/1 104 x 104 x 32 -> 104 x 104 x 128 20 0.099 BF 6 conte 5 4 1 conv 6 4 1 x 1/1 104 x 104 x 32 -> 104 x 104 x 128 20 0.099 BF 6 conte 5 4 1 conv 6 4 1 x 1/1 104 x 104 x 32 -> 104 x 104 x 128 20 0.099 BF 7 conv 6 4 1 x 1/1 104 x 104 x 32 -> 104 x 104 x 128 20 0.099 BF 	<pre>60 & & di_0514.kpmb :* Instant Mathem HAN AKTHEN IN EASHMANNE = + Hill + 2# 0 f (B) f dhe inse using our helper faction inder/ redictions.pr) </pre>
 ● ai_0514.ipynb ☆ 概率 編編 後現進面 描入 軟行階段 工具 認明 已結分析分配更 + 程式碼 + 文字 ■ This will output the image with the detections shown. The most recent detections are always saved to 'predictions.jpg' (a) Try out the examples below for yourself! ● # run darknet detection on test images 1./darknet detector test ofg/occo.data ofg/yolov4-tiny.efg yolov4-tiny.weights data/person.jpg ● CDB-version: 1080 (12000), euDN: 8.7.0, CDDN_HALF=1, GFU count: 1 CDNN_HALF=1 OpenCV version: 4.2.0 0 compute_camability = 750, cudm_half = 1, GFU count: 1 CDNN_HALF=1 OpenCV version: 4.2.0 1 cover 64 3 x 3/2 206 x 208 x 32 - 104 x 104 x 64 0.399 BF 2 cover 64 3 x 3/2 416 x 416 x 3 -> 208 x 208 x 32 0.075 BF 1 cover 64 3 x 3/1 104 x 104 x 32 -> 104 x 104 x 46 0.079 BF 3 route 2 1/2 -> 104 x 104 x 32 0.199 BF 5 cover 32 3 x 3/1 104 x 104 x 32 -> 104 x 104 x 64 0.079 BF 5 cover 64 1 x 1/1 104 x 104 x 32 -> 104 x 104 x 64 0.079 BF 5 cover 64 1 x 1/1 104 x 104 x 32 -> 104 x 104 x 64 0.079 BF 5 cover 64 1 x 1/1 104 x 104 x 32 -> 104 x 104 x 64 0.079 BF 5 cover 64 1 x 1/1 104 x 104 x 32 -> 104 x 104 x 64 0.079 BF 6 route 54 -> 104 x 104 x 32 -> 104 x 104 x 32 0.199 BF 7 cover 64 1 x 1/1 104 x 104 x 32 -> 104 x 104 x 64 0.079 BF 7 cover 64 1 x 1/1 104 x 104 x 32 -> 104 x 104 x 64 0.079 BF 7 cover 64 1 x 1/1 104 x 104 x 32 -> 104 x 104 x 64 0.079 BF 7 cover 64 1 x 1/1 104 x 104 x 125 -> 56 x 52 x 128 0.071 BF 10 cover 128 3 x 3/1 104 x 104 x 125 -> 56 x 52 x 128 0.071 BF 10 cover 108 2 cover 7 4 1 x 104 x 126 +> 106 x 106 x 125 +> 56 x 62 x 128 0.071 BF 10 cover 108 3 x 3/1 105 x 52 x 52 x 128 -> 56 x 52 x 128 0.071 BF 10 cover 108 2 x 52 x 128 -> 56 x 52 x 128 0.071 BF 10 cover 108 2 x 52 x 128 -> 56 x 52 x 128 0.071 BF 10 cover 108 2 x 52 x 128 -> 56 x 52 x 128 0.071 BF 10 cover 108 2 x 52 x 128 -> 56 x 52 x 128 0.071 BF 10 cover 108 4 x 124 x 125 -> 56 x 52 x 128	 A si_0514.kpmb :* ks was watawa HA wither I a was Laternamed + With + 2* 0 is 10 f alms insee using our helper faction inder's redictions.py')
 ● ai_0514.ipynb ☆ 描述 編編 後現遺画 描入 執行期段 工具 説明 已結存所引望更 + 程式碼 + 文字 ■ this will output the image with the detections shown. The most recent detections are always saved to 'predictions.jpg' (a) Try out the examples below for yourself! ● = run darknet detection on test images 1./darknet detector test cfg/coco.data cfg/yolov4-tiny.esights data/person.jpg ● = run darknet detection on test images 1./darknet detector test cfg/coco.data cfg/yolov4-tiny.esights data/person.jpg ● CDA-version: 1080 (12000), euDNN: 8.7.0, CDDNUHLF=1, GFU count: 1 CDNN.HALF=1 OpenCf Version: 4.2.0 0 : compute_camability = 750, cudm_half = 1, GFU: Tesla T4 net.optimized_memory = 0 min_batch = 1, batch = 1, time_steps = 1, train = 0 layer filters size/strd(dil) input output 0 Create cudm-handle 0 conv = 64 = 3 x 3/2 = 206 x 208 x 32 - 104 x 104 x 46 4.0.797 EF 3 route 2 = 3 x 3/2 = 104 x 104 x 104 x 42 - 104 x 104 x 46 4.0.797 EF 3 route 2 = 1/2 - 104 x 104 x 32 0.199 EF 5 conv = 24 = 1/2 - 104 x 104 x 32 0.199 EF 5 conv = 24 = 1/2 - 104 x 104 x 42 + 104 x 104 x 32 0.199 EF ● max = 2 + 2/2 = 104 x 104 x 12 + 5/2 x 5/2 x 128 0.001 BF 11 route 10 = 1/2 - 5/2 x 5/2 x 128 - 5/2 x 5/2 x 128 0.001 BF 11 route 10 = 1/2 - 5/2 x 5/2 x 128 - 5/2 x 5/2 x 128 0.001 BF 11 route 10 = 1/2 - 5/2 x 5/2 x 128 - 5/2 x 5/2 x 128 0.001 BF 11 route 10 = 1/2 - 5/2 x 5/2 x 64 0.199 EF 	<pre>A sl_0514.kpmb :* tax and kodem into a kodem into a taken texture texture</pre>

圖1~4. 確認可執行YOLOv4-tiny範例

0

107

101

四、使用Google Colab製作AI模型- 換自己的資料集1



107

101

😑 my_yolov4-tiny_test.cfg 🗵

4

14

24

Normal text file

[net]

3 batch=1

angle=0

16 hue=.1

*D:\HUB 8735\0514\my_yolov4-tiny_test.cfg - Notepad++

檔案(F) 編輯(E) 授尋(S) 檢視(V) 編碼(N) 語言(L) 設定(T) 工具(O) 巨集(

12 🖻 🖶 🖻 12 🎧 🛆 🖌 🌇 Thio Cia 🐜 🔍 🔍 🖽 🗔 🗌

🛆 ai 0514.ipynb 🖄 💿

檢視書面 插入 執行階段 工具 說明 已儲存所有變更

+ 程式碼 + 文字

CO

檔案 编輯

+ 程式碼 + 文字 本機 > 本機磁碟 (D:) > HUB 8735 > 0514 Х ⊟ 檔案 [35] Loaded: U. UUUU/I seconds # Testing v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: [21] Q 日野 0515 秒 subdivisions=1 1影片 ObjectDetectionCallback # Training {*x*} darkı 測試模型 📕 HUB 8735 🖈 📴 60mins.zip 6 #batch=64 →移動 90mins.zip 0514 7 #subdivisions=1 3rdparty 8735路徑.txt 8 width=416 JPEGImages 0515 ai 0514.ipvnb 9 height=416 backup [39] !cp -r /content/gdrive/MyDrive/data/results/* /content/ ai test 5 back: 0.84 50 back: 0 my_ai.data 10 channels=3 步驟2 build 📄 my_ai.names momentum=0.9 Cfg 手動上傳 my_yolov4-tiny_test.cfg 和測試照片test.jpg OneDrive - Pers my_yolov4-tiny.cfg 12 decay=0.0005 cmake my_yolov4-tiny_10000.weights 🕒 本機 💼 data my_yolov4-tiny_last_18mins.v saturation = 1.5# run darknet detection on test images 🗊 3D 物件 include my_yolov4-tiny_last_60mins.we exposure = 1.5/darknet detector test my_ai.data my_yolov4-tiny_ ➡ 下載 labels my_yolov4-tiny_last_90mins.weights 5 front: 0.78 🖹 文件 my_yolov4-tiny_test.cfg 💼 obj 5 back: 0.83 🎝 音樂 network_binary.nb learning rate=0.00261 results 🔲 桌面 🜆 test.jpg burn in=1000 my_yolov4-tiny_10000.we... ■ 圖片 my_yolov4-tiny_20000.we. ■ 影片 max batches = 2000200my_yolov4-tiny_last.weig.. 🐛 本機磁碟 (C:) policy=steps tmp.txt steps=1600000,1800000 _ 本機磁碟 (D:) scripts scales=.1,.1 1 front: 1.00 back: 0.87 ▲ 網路 src <> CMakeLists.txt ▲ 拖曳檔案以上傳至工作階段儲存空間 16 個項目 已選取 2 個項目 35.9 KB #weights reject freq=1001 #ema alpha=0.9998 54.44 GB 可用 #aguidiatant noint-1000 ✓ 0秒 完成時間:凌晨12:25 圖3. 成功辨識 圖1. 新增及修改.cfg 圖2. 上傳測試模型設定與測試照片 D 成功编譯

、使用Google Colab製作AI模型- 換自己的資料集2

📙 | 🛃 📜 🖛 | 0514

共用 榆視 ction With Colab.ipynb

報 工具 說明 已儲存所有變更

30

矿四、使用Google Colab製作AI模型-換自己的資料集2

😋 ai_0515.ipynb - Colaboratory 🗙 🕂	■ *D:\HUB 8/35\0515\my_yolov4-tiny.ctg - Notepad++ #完化, 結果(1) 把意(2) 控制(0) 活躍(0) 预言(1) 把意(2) 万葉(0)			
← → C ((個条(F) 編輯(E) 授發(S) ((1)((V) 編卷(N) 語言(E) 改定(1) 工具(O) 巨果(M) → □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □	(M,1)(K))7,97(F) (H,⊠(W)) (+ ▼ × 1]		
🔀 智慧監控管理系統 G Google 😝 Facebook 💶 YouTube 🌐 國立限	🔚 my_yolov4-tiny.cfg 🛛			
CO ▲ ai_0515.ipynb ☆ 檔案 編輯 檢視畫面 插入 執行階段 工具 說明 已儲存所引	210 batch_normalize=1 211 filters=512			
+ 程式碼 + 文字 ○ × [10] from socile collab import drive	212 size=3 213 stride=1 214 pad=1	$ers = 3^{*}(5 + len(classes))$	D\HUB 8735\0515\my_ai.data - Notepad++	
20 drive. mount (' <u>/content/gdrive</u> ')	215 activation=leaky			
<pre>{x} Mounted at /content/gdrive</pre>	216 217 [convolutional]	尋找 取代 在多個檔案中尋找 在專案中多個檔案中尋找 標記	150_{front}	1 classes = 8
<pre>16 [11] !cp -r /content/gdrive/MyDrive/data/* /content/da 16 19 10 10 10 10 10 10 10 10 10 10 10 10 10</pre>	218 size=1 219 stride=1	##\$\$P3音· y00 2次 12 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	3 10_front 4 10 back	3 valid = ./test.txt 4 names = ./my ai.name
 → 需額外操作 	220 pad=1 221 filters=39 222 activation=linear		5 5_front 6 5_back 7 1_front	5 backup = ./results 6 results = ./results 7 eval = coco
於此手動上傳,編輯過之: my_ai.data, my_ai.names, my_yo	223 224	 ✓ 須濃(0) 按尋模式 ✓ 透明 ● 一般 ● 視磁失去架點時逸時 	8 1_back	
● [] %cd <u>/content/darknet</u> 0 %ls	225 226 [yolo] 227 maak = 2.4 5	○返申(n, r, r, n, 10, x) ○一直保持35明 ○規則運算式(g) 「」包含接行:		
● [] !./darknet detector train my_ai.data my_yolov4-t 0 秒	227 mask $= 3,4,3$ 228 anchors $= 10,14, 23,27,$ 229 classes=8	, 37,58, 81,82, 135,169, 344,319	圖2. 修改.na	mes跟.data
☆ ▼ 測試模型	230 num=6 231 jitter=.3 232 scale x y = 1.05		The o	
[] !cp -r /content/gdrive/MyDrive/data/results/* /co 0	Normal text file	Ln : 221 Col : 11 Pos : 2,308 Unix (LF) UTF-8 INS		
日 9 在這裡輸入文字來搜 日 9 日 9 日 9 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1		へ भ		
	圖1. 修改.cfg			

1 O T

心内、使用Google Colab製作AI模型-換自己的資料集3



107

矿四、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\4.0.3build20230510\variants\common_nn_models 宜、以Arduino IDE將模型燒入晶片-燒入自己的AI模型



10 T

六、補充-其他案例手把手教學



i≘ 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AloT 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	labelImg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,適用於智能家庭, 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

